

2019 年度 修 士 論 文

家電独自判断による 家庭内電力デマンド制御法

指導教員 本島 邦行 教授

群馬大学大学院理工学府理工学専攻

電子情報・数理教育プログラム

修士 2 年 T181D072

水澤 一太

まえがき

本修士論文は 2 部構成となっており，第 1 部と第 2 部では内容が異なる。

第 1 部では，「家電独自判断による家庭内電力デマンド制御法」と題し，家庭内の総消費電力を制御するためのアルゴリズムを示す。

第 2 部では，「IoT を用いたインテリジェント農作物育成環境解析」と題し，きゅうりの栽培環境を最適化するための解析結果を示す。

なお，2019 年度修士論文公聴会では第 1 部の内容である，

「**家電独自判断による家庭内電力デマンド制御法**」について発表する予定である。

審査を担当してくださる小林 春夫 教授 ならびに 弓仲 康史 教授，査読のほどよろしくお願いいたします。

目次

<第 1 部>

| | |
|--|----|
| 1. 序論 | 1 |
| 2. システムの構成 | 2 |
| 2.1. システム設計における方針(3 つのコンセプト) | 2 |
| 2.2. 現状のシステムと本研究の方向性 | 3 |
| 3. 制御アルゴリズム | 4 |
| 3.1. 「起動時」と「稼働中」 | 4 |
| 3.2. 制御対象の分類 | 4 |
| 3.3. Time Control Type (TCT) | 5 |
| 3.4. Power Control Type (PCT) | 7 |
| 4. 評価方法 | 15 |
| 4.1. シミュレータ | 15 |
| 4.2. 実験 | 16 |
| 4.2.1. 模擬スマートメーター | 16 |
| 4.2.2. 模擬家電 | 19 |
| 5. 検証 | 22 |
| 5.2. PCT-A, PCT-B, TCT, 非対応家電 A | 25 |
| 5.3. PCT-A, PCT-B, TCT, 非対応家電 A, 非対応家電 B | 26 |
| 6. 結論 | 27 |

<第 2 部>

| | |
|-------------------------------|----|
| 1. 序論 | 28 |
| 2. 解析データ | 29 |
| 2.1. データの種類 | 29 |
| 3. 多変量解析 ^[2] | 30 |
| 3.1. 多変量解析の種類 | 30 |
| 3.2. 外的基準のある多変量解析 | 31 |
| 3.3. 外的基準のない多変量解析 | 32 |
| 3.4. その他の多変量解析 | 34 |
| 3.5. 多変量解析の手法選択フローチャート | 34 |

| | |
|---|----|
| 4. 重回帰分析 | 36 |
| 4.1. 目的変数 y | 36 |
| 4.2. 説明変数 x | 36 |
| 4.3. 回帰式の評価方法 ^[3] | 38 |
| 4.4. 説明変数パラメータの改良 | 39 |
| 4.4.1. 日出時刻と日没時刻を用いたパラメータの2分割 | 40 |
| 4.4.2. 一日の気温差 | 41 |
| 4.4.3. 適正温度時間長 | 41 |
| 4.5. 多重共線性の考慮 | 42 |
| 4.5.1. 多重共線性とは ^{[4],[6]} | 42 |
| 4.5.2. 多重共線性の確認 | 42 |
| 4.6. 日射量の考慮 | 47 |
| 4.7. 解析結果 | 49 |
| 4.8. 多重共線性の再検証 | 51 |
| 4.9. 日射量が「少ない」時の重回帰分析 | 52 |
| 5. 収穫量予測 | 53 |
| 5.1. 収穫量予測の手法 | 53 |
| 5.2. 収穫量予測の評価 | 54 |
| 6. 深層学習 ^{[6][7]} | 58 |
| 6.1. ニューラルネットワークの構造 | 58 |
| 6.2. RNN 学習における問題点 | 59 |
| 6.3. Truncated Backpropagation Through Time | 60 |
| 6.3.1. RNN とは | 60 |
| 6.3.2. Backpropagation Through Time とは | 61 |
| 6.3.3. Truncated BPTT とは | 62 |
| 6.4. 実装 | 63 |
| 6.4.1. 実装の流れ | 63 |
| 6.4.2. データセット | 64 |
| 6.4.3. ミニバッチの作成 | 65 |
| 6.4.4. RNN 順伝播 | 65 |
| 6.4.5. Affine 順伝播 | 66 |
| 6.4.6. Time Softmax with Loss 順伝播 | 67 |
| 6.4.7. Time Softmax with Loss 逆伝播 | 68 |
| 6.4.8. Affine 逆伝播 | 68 |
| 6.4.9. RNN 逆伝播 | 69 |
| 6.4.10. 最適化 Optimizer | 69 |

| | |
|----------------------|----|
| 6.5. 機械学習 | 70 |
| 6.5.1. 各種パラメータ | 70 |
| 6.5.2. 学習結果 | 71 |
| 7. 結論 | 73 |
| 謝辞 | 74 |
| 参考文献 | 75 |
| 研究業績 | 77 |

第一部

家電独自判断による 家庭内電力デマンド制御法

1. 序論

2011 年 3 月 11 日に発生した東日本大震災は、地震、津波といった直接被害だけでなく、間接被害も引き起こすことになった。その一例として、東京電力が管轄する範囲内での計画停電が挙げられる。これにより、対象となった地域では交通網や社会活動に大きな影響を受け、民衆はたいへん不便な生活を強いられることになった。これは電力の不足により発生したものであり、電力需要側が自主的に電力使用を削減できていればこのような事態は起こらなかったはずである^[1]。

さらに、政府は地球温暖化対策^[2]として次のような内容を平成 28 年に閣議決定している。

「住宅全体で省エネルギー・省 CO2 を促進するため、エネルギーの使用状況を表示し、空調や照明等の機器が最適な運転となることを促す住宅のエネルギー管理システム (HEMS) が 2030 年までにほぼ普及することを目指すとともに、家庭における電気の使用量が従来よりも詳細に計測でき、HEMS との連携等により電力使用量の見える化を促すスマートメーターの導入を進める。」

この決定に伴い、各電力会社は「スマートメーター^[3]」と呼ばれる電力量計を全家庭に設置することを取り急ぎ行っている。スマートメーターとは通信機器が搭載されており、検針や契約アンペアの変更を電力会社との通信によって簡単に行うことができる。加えて家庭内に存在する HEMS^[4] (Home Energy Management System) 機器とも通信ができ、30 分おきに指示数を送ることができる機器になっている。HEMS とは住宅エネルギー管理システムであり、中央制御装置である HEMS コントローラが、住宅内の電力センサから環境情報を収集し、家電製品を一括制御することで、住宅の電力需要をコントロールするシステムである。しかし、このようなシステムが既に構築されているにもかかわらず、広く普及していない。その原因^[5]は、システムを導入による家のリフォームや HEMS が対応できる家電への交換といった大きな出費が必要となってくるためである。

そこで我々は、現行の HEMS 機器に代わる使用電力削減システムを新たに構築し、HEMS が現在抱えている問題を改善していこうと考えた。また、この取り組みは電力削減に対応するだけでなく、今後普及していく再生可能エネルギーの不安定な電力供給にも対応できるはずである。

2. システムの構成

この章では、本研究におけるシステムの構成について示す。

2.1. システム設計における方針(3 つのコンセプト)

今日存在している HEMS などの家庭内電力制御システムは、スマートメーターやスマート家電が家庭内でネットワークを構築し、電力のモニタリングや電力製品の遠隔操作が行えるようになっている。しかし、家電製品の制御権限が HEMS コントローラに集中し、システム構成が複雑になってしまっている。そこで、我々の研究ではこれらの問題をふまえ、以下の 3 つをコンセプト^{[6][7]}として掲げ、これらの要件を満たしたシステム構築を目指す。

(1) 簡易性

従来のシステム^[8]は家電の制御権限が HEMS コントローラに集中しているため、システムが複雑化してしまっている。そこで本提案システムでは、システム内での通信をスマートメーターが各家電に対して送信するだけの、単純でトラフィック量の少ない一方向通信を用いる。また、受信側の家電に制御権限を与え、独自のアルゴリズムによって各々の家電が電力需要制御の必要性を判断する。

(2) 堅牢性

電力需要制御権限を家電に与えることから、制御を行う判断材料を正確に受け渡すことが必要になる。そこでスマートメーターから各家電に送信するデータを「家庭内で目指すべき総消費電力『目標上限値 P_t 』」, 「現在の総消費電力『測定値 P_m 』」の 2 数のみとし、最低限のデータ量にする。また目標上限値 P_t は、電力会社がそのときの発電量に応じて各家庭のスマートメーターへ電力ネットワークを経由して通知することを想定していく。

(3) 柔軟性

メーカーによる制御が行われていない家電については、本提案システムを組み込んで制御する。トップランナー方式が組み込まれている家電など、すでに制御が行われている家電については、本提案システムを組み込むかどうかの判断はメーカーに委ね、組み込める家電に関しては同様に本提案システムによる制御を行う。すなわち、家庭内の電力系統に、他のシステムに対応した家電製品や、通信制御機能を持たない家電製品（従来家電）が混在していても運用が可能であるシステム開発とする。

2.2. 現状のシステムと本研究の方向性

各家庭の電力供給元には，図 2.1 に示すようなスマートメーターと呼ばれるものが設置されている。スマートメーターとは，毎月の検針業務の自動化を可能にする電力量計であり，使用量を 30 分毎に計測して自動で電力会社に送信するものである。さらに，スマートメーターは HEMS に対応しているため，Wi-SUN（Wireless Smart Utility Network）と呼ばれる通信方式を標準搭載している^[9]。

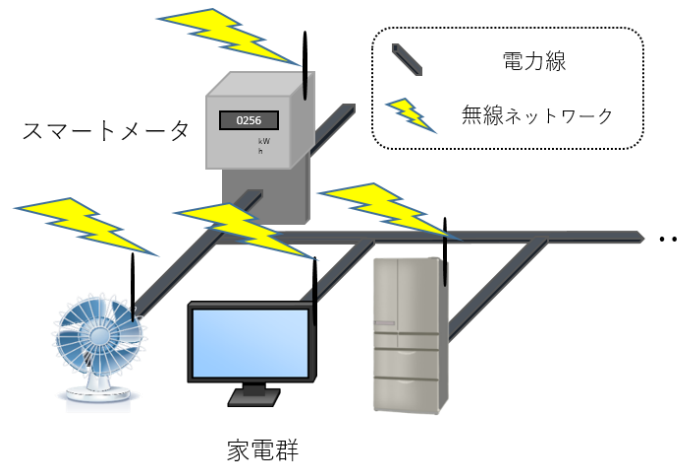


図 2.1：家庭内電力制御システム 概要

我々は，この通信方式 Wi-SUN を用いてスマートメーターから各家電に対して 3 秒に 1 回，現在の総消費電力である「測定値 P_m 」と家庭内で目指すべき総消費電力「目標上限値 P_t 」の 2 数を送信し，それを受信した家電は独自の判断によって制御され，電力を増減させることを目指す。これを反復していくことで家庭内の総消費電力は抑制されることになる。

3. 制御アルゴリズム

前の章では、本研究における方針とシステムの概要について述べてきた。本章では本研究の核となる、家電の制御アルゴリズムについて示していく。

3.1. 「起動時」と「稼働中」

本研究では家電製品の瞬間総消費電力を抑えることが目的である。しかし家電は、ユーザーの必要に応じて使われるものであるから、電力の制御範囲には限界がある。そこで、家電が制御されず自由に電力が使えるときの消費電力を P_{max} 、家電が正常に機能する最低限の消費電力を P_{min} とし、すべての家電にこの 2 数を設定する。そうすることで、各々の家電の消費電力は $P_{min} \leq P_{app} \leq P_{max}$ (P_{app} は家電そのものの消費電力) の範囲内で電力制御される。さらに効率的な電力削減を行うために、すべての家電は「起動時」と「稼働中」でアルゴリズムを分ける。その根拠は以下の通りである。

電力削減を強いられる要因には次の 3 点が挙げられる。

- (1) 家庭内で同時間に使用される家電が増えたとき
- (2) 電力会社からの目標上限値 P_t が引き下げられたとき
- (3) 個々の家電内で電力変動が生じたとき

これらの中で、電力削減を強いられる最も大きな原因となり得る事柄は(1)であると考えられる。つまり(1)の要因を減らすことが目標上限値内で電力を抑えるためには重要である。そこで、各家電に(1)の要因を減らすことに特化した「起動時」モードを設け、電力制御が行われた状態で家電が運転を開始することにした。そして「稼働中」では、(2)、(3)の要因に対して電力制御を行うアルゴリズムを実装する。先に述べた P_{min} や P_{max} はメーカーが独自に設定することができ、以降で記述するアルゴリズムにおいても製品に内蔵した形で実現することが可能である。

3.2. 制御対象の分類

家庭の電力需要制御を行うにあたり、最重要視するのはユーザーに不快感を与えないことである。不快感を与える要因には、(電力の削減による)家電の性能低下や、家電が起動しないことが考えられる。この 2 点を解決するために、まずは家電の運転終了を「待機可能な家電」と「待機不可能な家電」に分類した。待機可能な家電は起動時間を遅らせることで家庭内の消費電力のピーク時間を遅らせ、待機不可能な家電は先ほど述べた $P_{min} \leq P_{app} \leq$

P_{max} の範囲内で消費電力の削減を図る。

本研究では、待機可能な家電を Time Control Type（以降、TCT）、待機不可能な家電を Power Control Type（以降、PCT）とする。また、TCT と PCT に分類された家電の一例を表 3.1 に示す。

表 3.1 : 家電製品の分類

| タイプ名 | 制御対象 | 家電例 |
|------|--------|-----------------|
| TCT | 時間 | 乾燥機, 食洗器 |
| PCT | 電力, 時間 | 照明, 電子レンジ, エアコン |

3.3. Time Control Type (TCT)

乾燥機や食器洗浄機などの家電類は、家電そのものの運転が終了すると同時に、その中身をユーザーが必要とすることは少ない。すなわち、家電の使用優先度はそれほど高くないものと考えられる。そこで電力余裕があるときに、家電が運転を開始するようなアルゴリズムとした。

「起動時」ではユーザーの操作により TCT の家電に電源が入ると、家電はスマートメーターからの目標上限値 P_t と測定値 P_m の 2 数から以下のように使用可能余剰電力 P_{mar} を求め、自らの通常消費電力 P_{max} と比較する。

$$P_{mar} = P_t - P_m \quad (3.1)$$

(3.1) 式より算出された P_{mar} と、 P_{max} の大小関係から表 3.2 に従った使用電力制御を行う。その例として図 3.1-①のように時刻 t_0 で家電に電源が入ると、既に家電①と②が運転しており、そのときの P_{mar} (700W) では TCT の家電 (900W) が稼動するための必要電力 P_{max} に不足している。よって、TCT の家電は電源が入った状態で運転は開始せず、パケットを 3 秒おきに受信し、 $P_{max} \leq P_{mar}$ となるまで待機している。図 3.1-②中の時刻 t のように、 P_{mar} に十分な電力が確保できたとき、すなわち $P_{max} \leq P_{mar}$ となったときに、TCT の家電が運転を開始する。(図 3.1-③) 運転を開始すると「起動時」モードは終了し、「稼働中」モードへ移行する。

「稼働中」についてであるが、本 Type は電力余裕がある際に起動を開始するため、電力制御を必要としない。よって、運転開始後は制御対象から除外される。

表 3.2 : 起動時アルゴリズム (TCT)

| 関係式 | |
|------------------------|-----------------------------|
| $P_{max} \leq P_{mar}$ | 運転開始 |
| $P_{max} > P_{mar}$ | 次のパケットまで待機し, P_{mar} を再計算 |

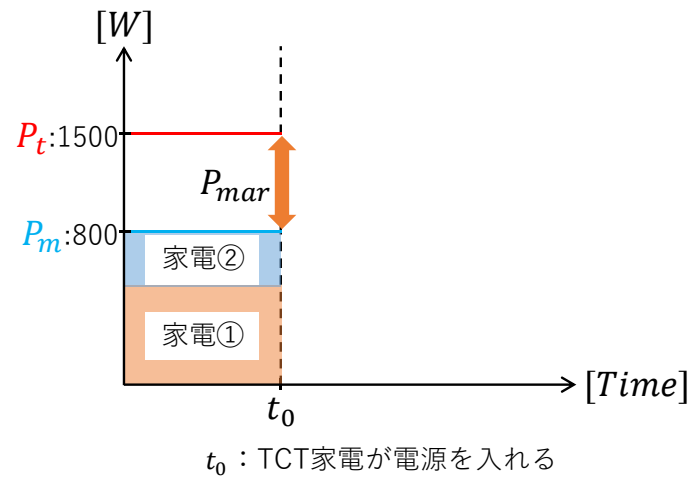


図 3.1 - ① : Time Control Type の制御方法①

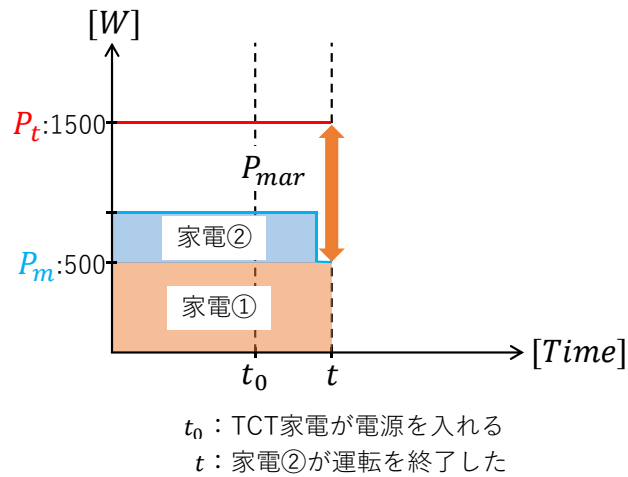


図 3.1 - ② : Time Control Type の制御方法②

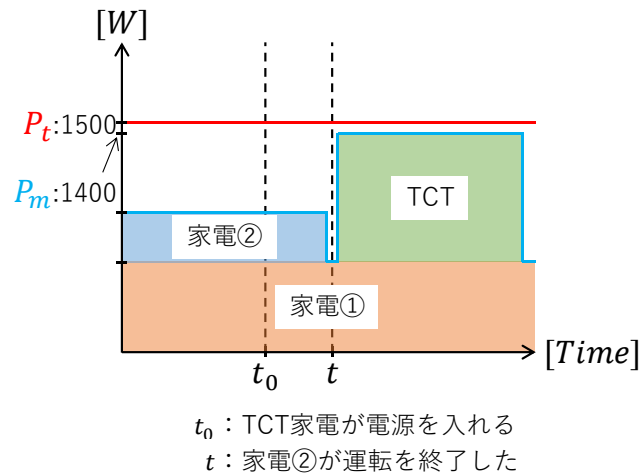


図 3.1 - ③ : Time Control Type の制御方法③

3.4. Power Control Type (PCT)

照明や電子レンジ、エアコンなどといった家電はユーザーが生活する環境に必要なものであり、運転終了を待望されているという特徴を持っている。そのため、TCT のように運転開始を遅らせるとユーザーはストレスを感じてしまう。そこで、電力制御によって瞬間消費電力の削減を図っていく。しかし、電力制御をすることは家電の性能低下に直結するため、ユーザーに不快感を与えないような工夫をする必要がある。

「起動時」は TCT と同様、図 3.1-①のように、ユーザーの操作により電源が入った時間 t_0 で、(3.1) 式で P_{mar} を求める。その値の大きさによって表 3.3 に示すように消費電力を制御しながら、運転を始める。

表 3.3 : 起動時アルゴリズム (PCT)

| 関係式 | 消費電力値 |
|-------------------------------|-----------|
| $P_{max} \leq P_{mar}$ | P_{max} |
| $P_{min} < P_{mar} < P_{max}$ | P_{mar} |
| $P_{mar} \leq P_{min}$ | P_{min} |

例えば表 3.4①に示すような PCT 家電が起動した場合、 $P_{min}(500) < P_{mar}(700) < P_{max}(800)$ となっているため、PCT 家電は 700[W]の電力で運転を開始する。②の場合では $P_{mar}(700) < P_{min}(800)$ であるため PCT 家電は 800[W]、③の場合では $P_{max}(600) < P_{mar}(700)$ であるため PCT 家電は 600[W]というようになる。②の場合のように P_{min} で運転してもなお、 P_{mar} を上回ってしまう場合には、他の家電に電力の融通を行ってもらうこととなる。(詳細は以降の稼働中のアルゴリズムで説明)

表 3.4 : PCT 家電の仕様 (説明用)

| | | |
|---|-----------|----------|
| ① | P_{max} | 800 [W] |
| | P_{min} | 500 [W] |
| ② | P_{max} | 1000 [W] |
| | P_{min} | 800 [W] |
| ③ | P_{max} | 600 [W] |
| | P_{min} | 450 [W] |

そして運転が始まると同時に家電は「起動時」モードから「稼働中」モードへ移行する。

「稼働中」ではその名の通り、家電が任務を遂行している中での電力制御となる。PCT 家電は定義の部分を参照してもらえればわかるが、運転終了(任務完了)を待望されている家電群の総称としている。そのため、稼働中での電力制御を大雑把に行ってしまうとユーザーの不快感に直結するため、細心の注意を払わなければならない。

そこで、ウェーバーの法則^[10]と呼ばれる技術をアルゴリズムに組み込んだ。ウェーバーの法則を一言で表せば、「我々の体を感じる感覚量は実際に受ける物理量の対数に比例する」というものである。内容をより理解するために具体的な数値を扱って示す。

例えば、1.0 [kg]の荷物を持っていたとする。ここに 10.0 [g]ずつ重りを増やしていき、ちょうど 50.0 [g]増やしたときにその増加に気づいたならば、その差は 50 [g]である。同様にして、2.0 [kg]の荷物で実験を行うと、ちょうど 100 [g]の増加まではその変化に気づかない。この性質がウェーバーの法則である。これを数式化すれば(3.2)式のようなになる。

$$K = \frac{\Delta R}{R} \quad (3.2)$$

R は刺激量、 ΔR は弁別閾と呼ばれ、上述した例の 100 [g]や 200 [g]がそれを指す。 K はウェーバー比と呼ばれるものであり、定数となる。

(3.2)式についてより深く見ていく。先ほど示した例を表で表してみると、表 3.5 のようになる。

表 3.5 : 刺激量と弁別閾の関係

| | | | | | | | | | |
|-----------------------|------|------|-----|------|------|-----|------|------|-----|
| 刺激量 R [kg] | 1.00 | 1.01 | ... | 1.05 | 1.06 | ... | 1.10 | 1.11 | ... |
| 弁別閾 ΔR [g] | 50.0 | 50.5 | ... | 52.5 | 53.0 | ... | 55.0 | 55.5 | ... |

ここで新たに感覚量というパラメータを導入する。このパラメータは刺激量 R が 1.0[kg]の時の感覚量を 1 とし、人間が刺激量の変化に気づいた時、すなわち $R + \Delta R$ 以上となった時、感覚量を 1 増やす。これを繰り返し、表にしたのが以下の表 3.6 である。

表 3.6 : 感覚量の定義

| | | | | | | | | | | |
|-----------------------|------|------|------|------|------|------|------|-----|------|-----|
| 刺激量 R [kg] | 1.00 | 1.01 | 1.02 | 1.03 | 1.04 | 1.05 | 1.06 | ... | 1.11 | ... |
| 弁別閾 ΔR [g] | 50.0 | 50.5 | 51.0 | 51.5 | 52.0 | 52.5 | 53.0 | ... | 55.5 | ... |
| 感覚量 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | ... | 3 | ... |

ここで表 3.6 中の感覚量が切り替わった部分のみを抜き出し数式化すると、次の(3.3)式になる。

$$R_{n+1} = (1 + K) \cdot R_n \quad (3.2)$$

(R_n : 感覚量 n の時の刺激量, K : ウェーバー比)

この式を変形すると次のような一般式が導出される。

$$R_n = R_1(1 + K)^{n-1} \quad (3.3)$$

以上のことから、ウェーバーの法則である「我々の体を感じる感覚量は実際に受ける物理量の対数に比例している」ことがわかる。

この式を用いることで、消費電力 P_{app} で運転している家電は、 $(1 - K) \cdot P_{app} \leq P_{app} \leq (1 + K) \cdot P_{app}$ の範囲内であれば電力制御を行っても人は気付きにくい。このように、ユーザーが気付くか気付かないかの境界を算出し、この範囲での電力制御をする。そして少し

時間を置き、ユーザーがその変化に慣れてから再び算出し、電力制御をする。これを繰り返すことで、ユーザーの不快感を最低限に抑えつつ電力制御を行うことができる。

図 3.2 にウェーバーの法則をアルゴリズムに組み込んだ家電と組み込んでいない家電の電力変動時間変化を示した。図からわかるようにウェーバーの法則を組み込んだ家電は階段状に少量ずつ電力削減を繰り返すことによって、ユーザーの不快感を抑えながら電力変動させている。しかし、ウェーバーの法則を組み込まない家電は目標上限値 P_t の変化に追従して、即座に電力削減を行ってしまうことからユーザーにより強い不快感を与えていると考えられる。

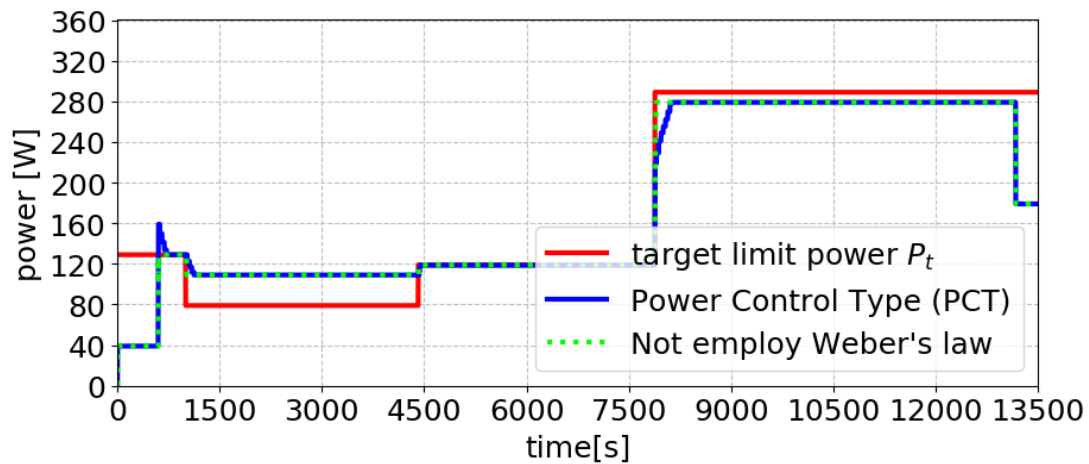


図 3.2 : ウェーバー則の必要性

「稼働中」では起動時アルゴリズム終了後も継続して、3 秒に 1 度の頻度で、目標上限値 P_t と測定値 P_m の 2 数を受信する。そこから(3.1) 式を用いることで P_{mar} を算出し、表 3.7 のような制御を行う。

表 3.7：制御アルゴリズム（PCT）

| 関係式 | | | 消費電力 |
|---------------|-----------------------------------|---|---------------------|
| $P_{mar} < 0$ | $-P_{mar} \geq K \cdot P_{app}^*$ | $\cap \frac{(1-K)P_{app} \geq P_{min}}{(1-K)P_{app} < P_{min}}$ | $(1-K)P_{app}$ |
| | | \cap | P_{min} |
| | $-P_{mar} < K \cdot P_{app}^*$ | $\cap \frac{P_{app} + P_{mar} \geq P_{min}}{P_{app} + P_{mar} < P_{min}}$ | $P_{app} + P_{mar}$ |
| | | \cap | P_{min} |
| $P_{mar} > 0$ | $P_{mar} \geq K \cdot P_{app}^*$ | $\cap \frac{(1+K)P_{app} \leq P_{max}}{(1+K)P_{app} > P_{max}}$ | $(1+K)P_{app}$ |
| | | \cap | P_{max} |
| | $P_{mar} < K \cdot P_{app}^*$ | $\cap \frac{P_{app} + P_{mar} < P_{max}}{P_{app} + P_{mar} \geq P_{max}}$ | $P_{app} + P_{mar}$ |
| | | \cap | P_{max} |

* P_{app} は現在の家電の消費電力

例えば，表 3.4①のような仕様の PCT 家電があったことを想定する。図 3.3-①のように時刻 t_0 で PCT 家電に電源が入ると，最低消費電力 $P_{min} = 500$ [W] で運転を開始する。（起動時アルゴリズム）ここである時刻 t で家電②が運転を終了し，目標上限値 P_t と測定値 P_m の差（ $P_{mar} > 0$ ）に余裕ができると，PCT 家電は図 3.3-②のように少しずつ電力制限を緩和させ，目標上限値 P_t を越さない範囲で家電自身の最高パフォーマンスに切り替えていく。

この例では，家庭内の家電が減少した時の電力規制緩和アルゴリズムを示したが，家庭内の家電が増加し，電力を削減していきたい場合も同じ工程を踏むこととなる。

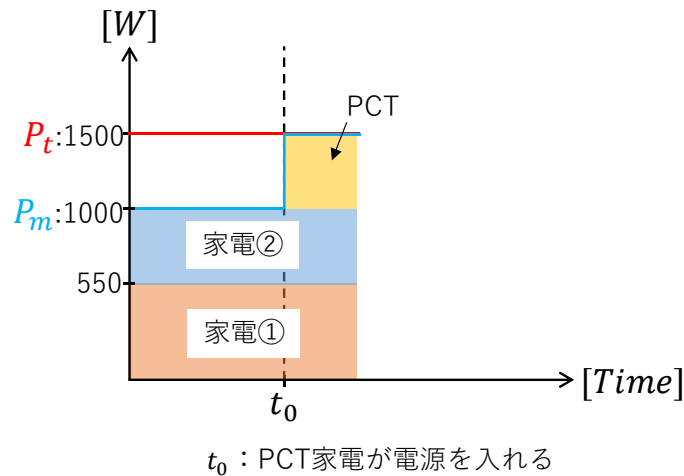


図 3.3 - ①：稼働中アルゴリズム（PCT）

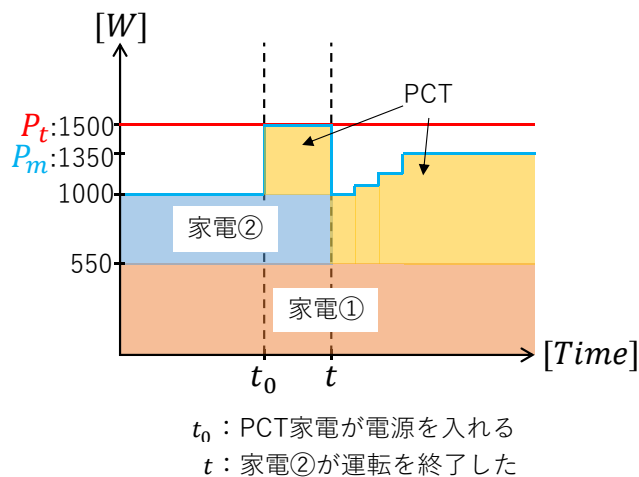


図 3.3 - ② : 稼働中アルゴリズム (PCT)

さらに、このタイプには電子レンジのようにユーザーによって運転する時間を設定されて開始する家電が存在し、表 3.7 のように電力を制御されてしまっは制御されたときの運転後と無制御時の運転後の出来に差が生まれてしまう。そこで、電力を削減した分運転時間を延長し、最終的に家電が運転時に使用した消費電力量を制御されたときとされていないときとで同等にするようにする。これを運転中は常に繰り返すことで、電力の時間変動に対応していく。

例えば、表 3.8 のような仕様の PCT 家電があったとする。

表 3.8 : PCT 家電の仕様②

| | |
|-------------|-----------|
| P_{max} | 800 [W] |
| P_{min} | 500 [W] |
| ユーザーによる設定時間 | 300 [sec] |

図 3.4-①のように時刻 t_0 で PCT 家電に電源が入ると、本来であれば $P_{max} = 800$ [W]で運転開始したいところを P_{max} の計算から、 $P_{app} = 500$ [W]で運転を開始していく。図 3.4-②のようにある時刻 t で家電②が運転を終了し、PCT は電力規制緩和を行いながら運転を続けていく。ここで、ユーザーによる時間設定が行われている場合と行われていない場合の違いは図 3.4-①、②に記された緑色の□枠を見てもらえれば理解しやすいと思われるが、表

3.8 の仕様によると運転開始 t_0 から、300 秒経過した時刻 ($t_0 + 300$) が本来の終了時刻である。しかし、電力制御が行われた場合と行われていない場合の運転にムラがあれば、ユーザーの不快感に直結しかねないため、運転時間を延長し消費電力量は一定となるように設計した。

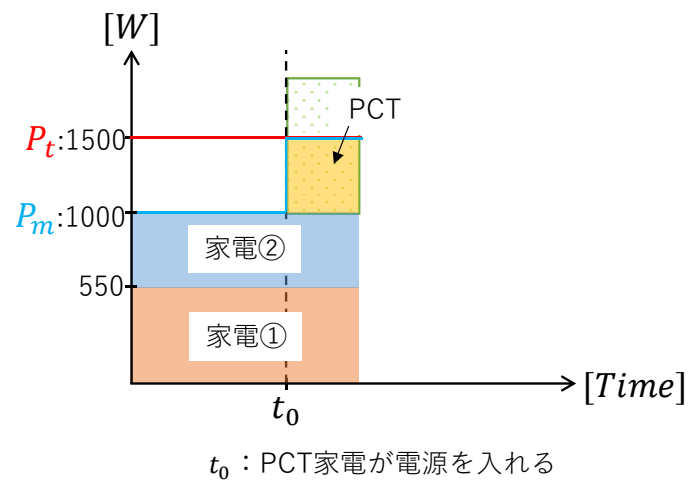


図 3.4 - ① : 稼働中アルゴリズム (PCT)

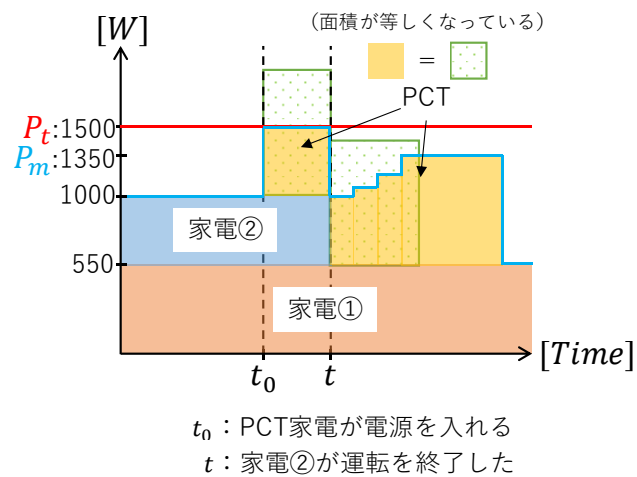


図 3.4 - ② : 稼働中アルゴリズム (PCT)

ここまで TCT と PCT のアルゴリズムを示してきたが、この 2 つの違いを表 3.9 に示した。

表 3.9 : TCT と PCT の違い

| | | TCT | PCT |
|---------|----|-----|-----|
| 制御対象 | 電力 | - | ○ |
| | 時間 | ○ | ○ |
| 電力抑制 | | - | ○ |
| 電力規制緩和 | | - | ○ |
| 運転開始の遅延 | | ○ | - |
| 運転時間の延長 | | - | ○ |

4. 評価方法

3 章には、私が考案した制御アルゴリズムについて示してきた。この制御アルゴリズムの有用性を検証するために、シミュレーションと実験の 2 通りを用いた。以下にその方法を示していく。

4.1. シミュレータ

開発したアルゴリズムを評価するためには、アルゴリズムを搭載した家電を開発し、導入したモデルハウスに人間が居住することが望ましい。しかし、このような評価は簡単ではない。そこで本研究では、計算機を用いた家庭内マイクログリッドシミュレータ^[11]を開発し、評価を行った。各家庭には数台～数十台もの家電製品があり、各家電は目標上限値 P_t と測定値 P_m を受信し、自己判断で消費電力を制御する。すなわち、異なる家電同士が並行して制御アルゴリズムを演算して使用電力を制御している。こうした状況は実際の環境では当たり前であるが、通常のプログラミングではシミュレートするのが大変困難である。したがって、本研究のシミュレータでは各家電の電力制御処理を個別のタスク上に実装して、マルチタスクで処理を行うことにした。そうすることで実際の環境により近い結果を得ることができる。

また、実験でアルゴリズムの検証を行うためには、実時間を用いて評価することしかできない。(1 日分の家電の動きを実験するのであれば、1 日を使わなければならない。) しかし、シミュレータは高速化に対応しており、1000 倍速まで評価に支障がないということが高橋昌史氏の修士論文に記されている。すなわち、1 日分の家電の動きを実験するのに 1 分 30 秒で済むことになる。

さらに、実験では困難になる太陽光発電システムによる発電の考慮や、蓄電システムによる電力の備蓄を考慮する場合にも対応ができるものとなっている。

4.2. 実験

本提案のアルゴリズムを検証するためにシミュレータのみならず実験も行う。しかし、実際に存在するスマートメーター及び家電製品にアルゴリズムを実装することが困難である為、「模擬スマートメーター」と「模擬家電」の 2 種類の装置から構成される模擬マイクログリッドの実験装置を開発した。以下に各装置の性能を示していく。

4.2.1. 模擬スマートメーター

本研究で製作した模擬スマートメーター（図 4.1）は通過電力測定を行い、測定した電力情報 P_m と目標上限値 P_t の送信を行う装置である。この装置は大きく 3 つの機構から構成される。

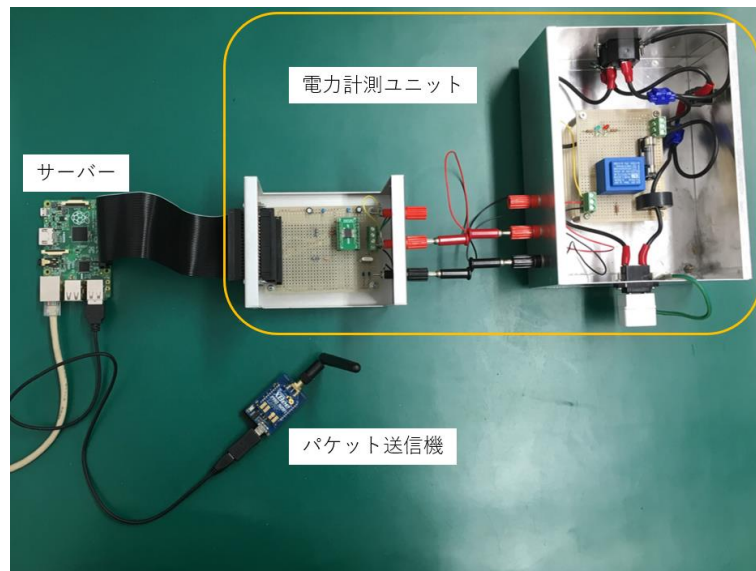


図 4.1 : 模擬スマートメーター

4.2.1.1. 電力計測ユニット

図 4.1 中、黄色の枠で囲われた 2 つの筐体が電力計測ユニットにあたる。このうち右側の筐体は電流電圧検出回路、左側の筐体内は電力の計測器となっている。電力計測には ADE7753 と呼ばれる電子計測専用の IC を用いた。(図 4.2) これはアナログデバイス社が提供しており、単相交流対応で 1ch の給電線の電圧・電流と移送関係を計測することができる。周辺のセンサデバイスから取得した情報を IC で処理し、皮相電力、有効電力、無効電力などを自動的に計算することができる。本研究では有効電力のみを取得し、SPI 通信によってサーバー側へ伝送されることとなる。

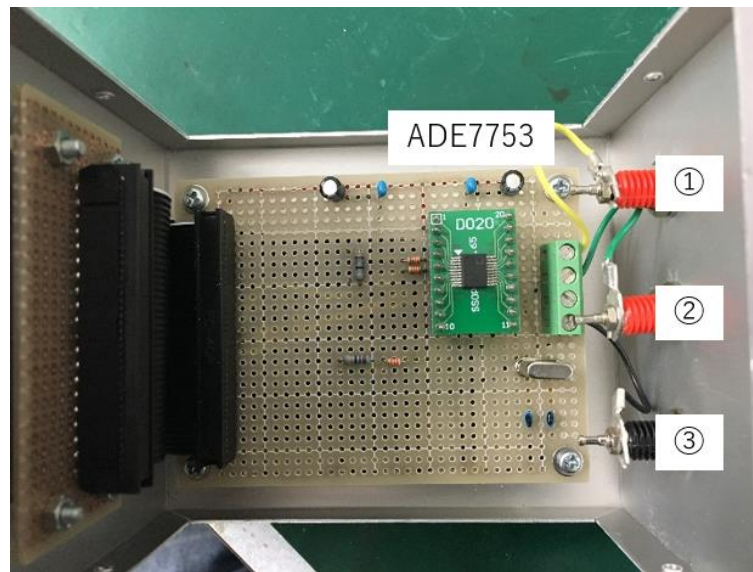


図 4.2 : ADE7753

図 4.2 中①は電圧を読み取る時の端子，②は電流を読み取る時の端子，③は GND 端子となっている。我々の研究では電流を読み取ることによって，消費電力の算出を行っているため，主に②と③端子を用いて実験を行っていく。

4.2.1.2. サーバー

サーバーには Raspberry Pi を利用している。Raspberry Pi は内蔵ハードディスクを搭載しない代わりに電源や SD カードストレージを装着することによって使用できるハードウェアとなっている。この Raspberry Pi が，電力計測ユニットによって算出された測定値 P_m と目標上限値 P_t をひとまとめにしたパケットを送信機へ送る。目標上限値 P_t は本来，電力会社が送信してくることを想定しているが，本実験ではサーバーに PC で作成したプログラムを実行させることで環境を整えた。



図 4.3 : Raspberry Pi 3 Model B

4.2.1.3. パケット送信機

近年流通している HEMS 機器は、スマートメーターから Wi-SUN と呼ばれる無線通信規格によって情報のやり取りを行っている。この Wi-SUN と呼ばれる通信方式はスマートメーターに標準搭載されており、電力会社がスマートメーターを全家庭に取り付けようと設置を進めている。(2019 年 7 月時点で普及率 79%^[12]) Wi-SUN は日本において特定小電力無線と呼ばれる 920MHz 帯で使用され、Wi-Fi などと比べると通信速度は遅いものの、通信距離は長く、障害物にも強く、さらには低消費電力という利点を兼ね備えている。さらに、ネットワーク容量も大きいため、我々の目的に最適となる通信規格であることは間違いない。(表 4.1) しかし Wi-SUN モジュールは高価である為、本研究の実験においては Wi-SUN の代替規格を用いることとした。代替通信規格として譲れない条件は①低消費電力の通信規格であり、②ある程度大きなネットワーク容量を備えているという点である。これらを兼ねている通信規格は ZigBee^{[13],[14]}ということになる。よって ZigBee を採用することとした。

上述した ZigBee 規格を搭載した無線通信モジュール XBee を図 4.4 に示す。

表 4.1 : ワイヤレス規格の比較

| IEEE 規格 | 802.15.11b | 802.15.1 | 802.15.4 | 802.15.4g |
|--------------|------------|-----------|-----------|----------------------|
| マーケット名 | Wi-Fi | Bluetooth | ZigBee | Wi-SUN |
| 周波数帯 | 2.4GHz | 2.4GHz | 2.4GHz | 920MHz |
| 変調方式 | CCK, PBCC | GFSK | O-QPSK | FSK, OFDM, O-QPSK |
| 拡張方式 | DSSS | FHSS | DSSS | 2GFSK/4GFSK |
| 通信距離 | 100m | 10m | 30m | 500m |
| 伝搬速度 | 600Mbps | 1Mbps | 250Kbps | 400Kbps |
| 送信出力 | 10mW/MHz | 2.5mW | 1mW | 20mW |
| ネットワーク 容量 | 32 ノード | 7 ノード | 65528 ノード | 65528 ノード |

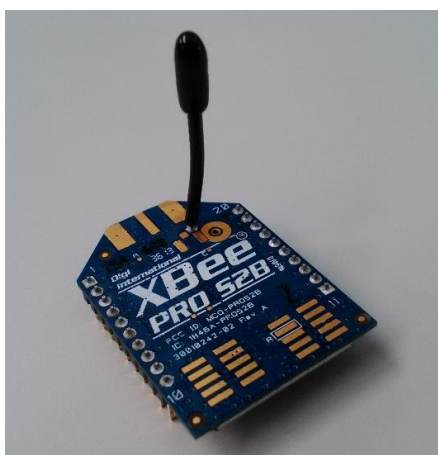


図 4.4 : XBee PRO S2B

4.2.2. 模擬家電

模擬家電（図 4.5）とは XBee 無線機を搭載し、運転を独自にコントロールできる電力消費装置である。対象となる家電製品を想定した制御アルゴリズムを実装することで、様々な家電を擬似的に表現することが可能である。模擬家電においても、3 つの機構から構成されている。

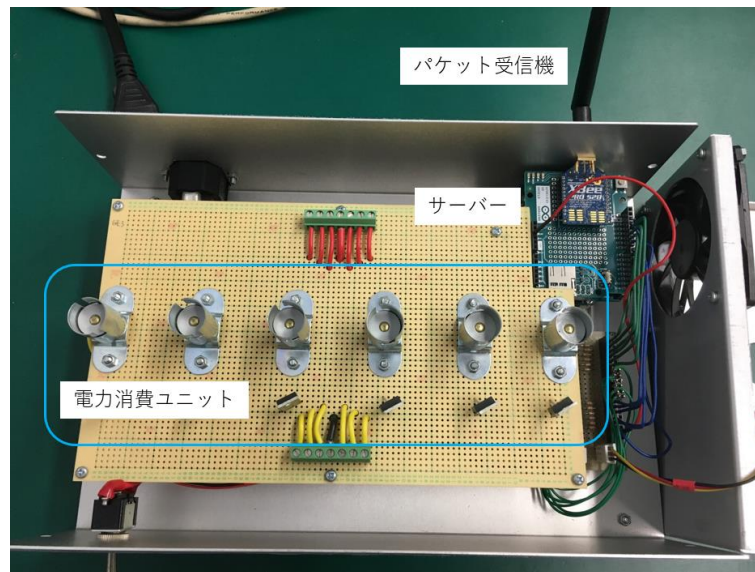


図 4.5 : 模擬家電

4.2.2.1. パケット受信機

パケット受信機も送信機同様，XBee を適用している。

4.2.2.2. 電力消費ユニット

電力消費ユニットには半導体デバイスである MOSFET と車載用の白熱電球を採用している。MOSFET のゲート端子へサーバーから信号を送ることによって PWM 制御を行い、電球が消費する電力の制御を行っている。本研究では電力消費ユニットに 23W 電球を 6 個設置しているため、最大で 138W の電力消費を行うことができる。

4.2.2.3. サーバー

サーバーは、パケット受信機である XBee が受信した目標上限値 P_t と測定値 P_m を用いて、独自のアルゴリズムを実行する。そのアルゴリズムにより決定した命令で電力消費ユニット運転し、消費電力を制御する。前述したように、消費電力を調整するために MOSFET を PWM 制御している。そこで PWM 制御が行える Arduino を採用した。

Arduino とは「AVR マイコン、入出力ポートを備えた基板で、Arduino 言語とその統合開発環境から構成されるシステム」のことである。このマイコンは Arduino 言語と Arduino 開発環境によってプログラミングを行うことができ、Arduino プロジェクトは、独立した動作をする。図 4.6 には模擬家電に搭載した Arduino UNO (Arduino の 1 種) を示す。Arduino にはさまざまな種類が存在しており、用途によって種類を使い分けていく。

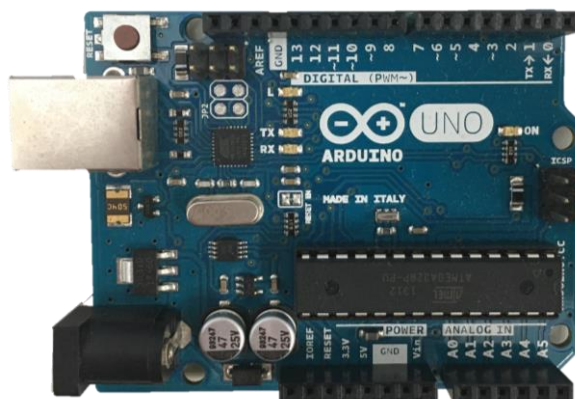


図 4.6 : Arduino UNO

5. 検証

3 章で示してきた，制御アルゴリズムの性能を評価する。実証試験においては環境設定として，目標上限値 P_t を表 5.1 のように設定した。しかし，実験において模擬家電 1 台に対して約 13W がベース電力として消費されてしまう。（模擬家電内で交流電源を直流に変換するための AD コンバータによる損失である。）そのため，実験に用いる模擬家電の台数 \times 13W（模擬家電 3 台の場合には 39W）を目標上限値 P_t に上乗せして設定した。

表 5.1：目標上限値 P_t の設定

| 経過時間 | 目標上限値 P_t [W] | 経過時間 | 目標上限値 P_t [W] |
|-------------|-----------------|-------------|-----------------|
| 0 - 4000 | 110 | 6000 - 6500 | 130 |
| 4000 - 4500 | 100 | 6500 - 7500 | 170 |
| 4500 - 5200 | 80 | 7500 - | 250 |
| 5200 - 6000 | 70 | | |

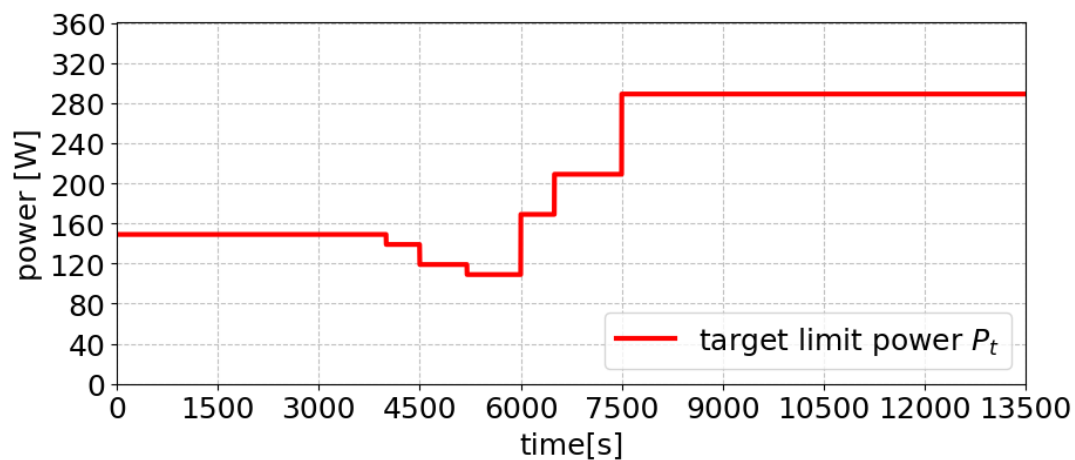


図 5.1：目標上限値 P_t の推移（家電 3 台の場合）

さらに，TCT，PCT，非対応家電の使用は次のように設定した。（表 5.2）

表 5.2：模擬家電の仕様

| | TCT | PCT - A | PCT - B |
|---------------|------|---------|---------|
| P_{max} [W] | 100 | 80 | 60 |
| P_{min} [W] | 0 | 30 | 40 |
| 起動時間 [sec] | 3600 | 600 | 600 |
| 稼働時間 [sec] | 5400 | 36000 | 36000 |
| Weber 比 | - | 0.10 | 0.10 |

| | 非対応家電 - A | 非対応家電 - B |
|---------------|------------|------------|
| P_{max} [W] | 70 | 40 |
| P_{min} [W] | - | - |
| 起動時間 [sec] | 1200, 6500 | 1000, 9600 |
| 稼働時間 [sec] | 1200 | 600 |

TCT の家電は実験開始から 3600 秒後にユーザーが電源を入れたと仮定する。このときの P_{max} は 100 [W] で $P_{max} \geq P_{max}$ となってから TCT は 5400 秒間運転する。PCT の家電として、 P_{max} と P_{min} が異なる A, B 2 つのタイプを用意した。 P_{max} と P_{min} 以外は共通であり、①実験開始から 600 秒のときにユーザーが電源を入れ、②10 時間 (36000 秒) ほど運転するとして、③ウェーバー比は 0.1 と仮定した。また、タイプ A, B の消費電力は $30 \leq P_{typeA} \leq 80$ [W], $40 \leq P_{typeB} \leq 60$ [W] である。さらに非対応家電も A, B の 2 タイプ準備した。A タイプは実験開始後 1200 秒, 6500 秒の 2 回 P_{max} を 70W として 20 分 (1200 秒) 間, B タイプは実験開始後 1000 秒, 9600 秒の 2 回 P_{max} を 40W として 10 分 (600 秒) 間, 家庭内の電力使用状況を見逃した運転を行うとした。

以上に記述した家電の設定で、複数の家電が組み合わさった場合にはどのような結果を以下に示す。

5.1. PCT-A, PCT-B, TCT

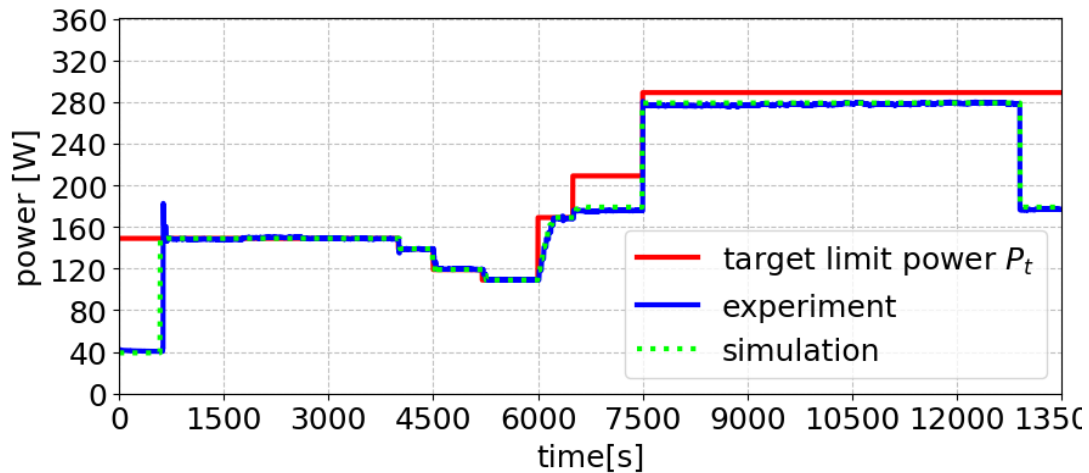


図 5.2 : PCT-A, PCT-B, TCT の 3 台における消費電力の推移

PCT-A, B と TCT の 3 台を運転させた場合の結果が図 5.2 である。開始 600 秒で PCT-A と PCT-B が起動した時、実験結果とシミュレーション結果で差異が見られる。これは 2 台の家電がパケットを受け取ってから制御するまでの時間を乱数で設定しており、この実験では偶然に乱数が一致してしまったことで 2 台の家電が同時に起動を開始しまったことにより、一時的に目標上限値 P_t を上回ってしまった。開始 5400 秒でユーザーが電源を入れた TCT の家電は 2100 秒 (35 分) 間待機し、開始 7500 秒で運転を開始した。シミュレーション結果と実験結果は開始 600 秒付近を除き、ほぼ一致した波形となった。

5.2. PCT-A, PCT-B, TCT, 非対応家電 A

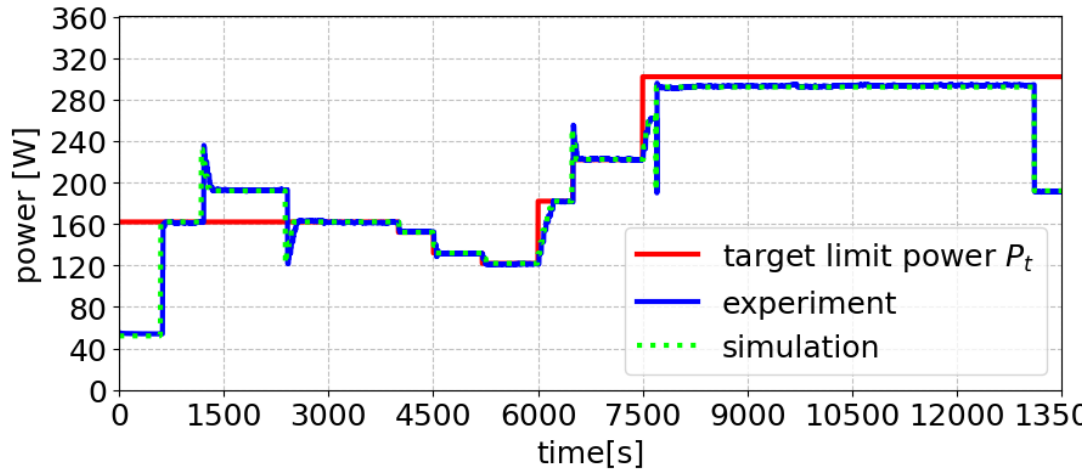


図 5.3 : PCT-A, PCT-B, TCT, 非対応家電 A の 4 台における消費電力の推移

5.1.1 で組み合わせた 3 台に非対応家電 A を加えて運転した場合の結果が図 5.3 である。5.1.1 の条件時では開始 600 秒で一時的に目標上限値 P_t を超えるような起動になってしまったが、この実験では乱数により、目標上限値 P_t を超えることは無かった。開始 1200 秒では非対応家電が運転を開始し、対応家電である PCT2 台が電力削減制御を行い、 P_t 以下まで引き下げることができなかった。2400 秒になると非対応家電が運転を終了し電力余裕ができると、PCT2 台は電力規制緩和制御を行い、 P_t 以下で追従した。こちらも実験結果とシミュレーション結果はほぼ一致した。

5.3. PCT-A, PCT-B, TCT, 非対応家電 A, 非対応家電 B

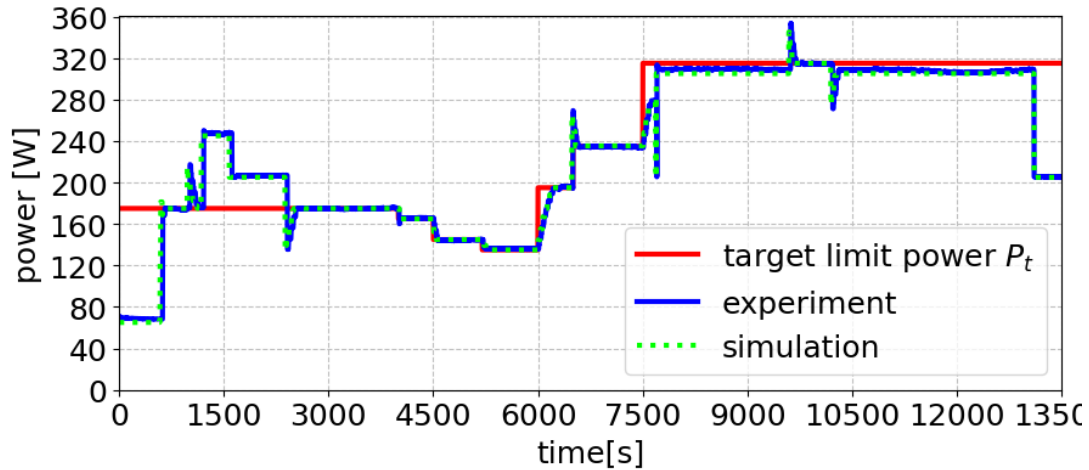


図 5.4 : PCT-A, PCT-B, TCT, 非対応家電 A, B の 5 台における消費電力の推移

PCT-A, B, TCT, 非対応家電 A の 4 台に非対応家電 B を加え, 5 台で運転した場合の結果が図 5.4 である。非対応家電が複数台に増えても, 他の対応家電が電力を融通しながら目標上限値である, P_t に追従することができた。

この実験から, 本提案システムを搭載した家電 3 台 (総消費電力 240W), 非対応家電 2 台 (総消費電力 110W) のもとでは十分デマンドカットが見込める。ただし P_{max} の大きさや, P_{max} と P_{min} の幅など, 様々な条件がデマンドカットの効果に影響を及ぼすこととなる。すなわち, 本システムに対応する家電が多いほど電力の制御範囲広げることができるため, デマンドカットの効果は大きくなる。

6. 結論

本研究では従来までの電力制御システムである HEMS の課題を改善し、簡易性・堅牢性・柔軟性をコンセプトとした新たなアルゴリズムを考案した。そのアルゴリズムを制御する手段として、スマートメーターから家電に対して目標上限値 P_t と測定値 P_m をひとまとめとしたパケットを一方向通信として送信し、各家電が受信したパケットに基づいて独自の判断で制御方法を決めるシステムとした。制御方法として、家電が電力削減を強いられる要因から「起動時」と「稼働中」の二つにアルゴリズムを分け、その要因ごとの制御方法を区別化した。次に家電の運転終了をユーザーが待望しているか否かによって家電を分類し、前者を Power Control Type、後者を Time Control Type と名付け、それぞれのアルゴリズムを示した。特に Power Control Type では、人間が持つ感覚と実際の物理量の違いを利用して、「人間に気付かれにくい電力制御」を提案した。最後に、提案したアルゴリズムをシミュレータと実験の両方から検証し、アルゴリズム通りの結果が得られたことから、このアルゴリズムを用いることで家庭内の瞬間総消費電力を抑えられることが明らかとなった。

今後の課題として、発電装置や蓄電装置など様々な因子が加わってきたときのアルゴリズムの考案や、家庭内の電力制御に固執せず、工場や農業の工場栽培に応用していくことを考えている。

第二部

IoT を用いた インテリジェント農作物育成環境解析

1. 序論

近年、日本の農業は衰退産業と言われている。図 1.1 をみると分かるように農業就業者の高齢化や就業者数の減少が大変深刻になってきているのは一目瞭然である。しかし、こういった事態を引き起こしているのには原因がある。それは、農作物の育成方法にあると筆者は考えている。現在の育成方法は生産者の経験と勘に依存しており、農作物を育成するためのマニュアルが存在しない。そのため、ひとたび農業に参入したとしても、農作物育成を効率的に行えるようになるのには長い年月を要してしまう。これが、新たな就業者を加わり辛くさせる原因になっている。この状況を変えるには、経験や勘に依存しない合理的なマニュアルが必要不可欠であると考えられる。

そこで我々の研究では、JA 全農ぐんま様よりきゅうりの栽培環境データを提供いただき、それらを多変量解析および人工知能（AI）を用いて統計的に解析していくことで、農作物育成において必要な環境因子を見つける。加えて、それらの環境因子の最適値を求めることで高収量化を目指す。このように、栽培環境を定量的に評価することで経験や勘に依存しない合理的な作物育成環境を構築していく。

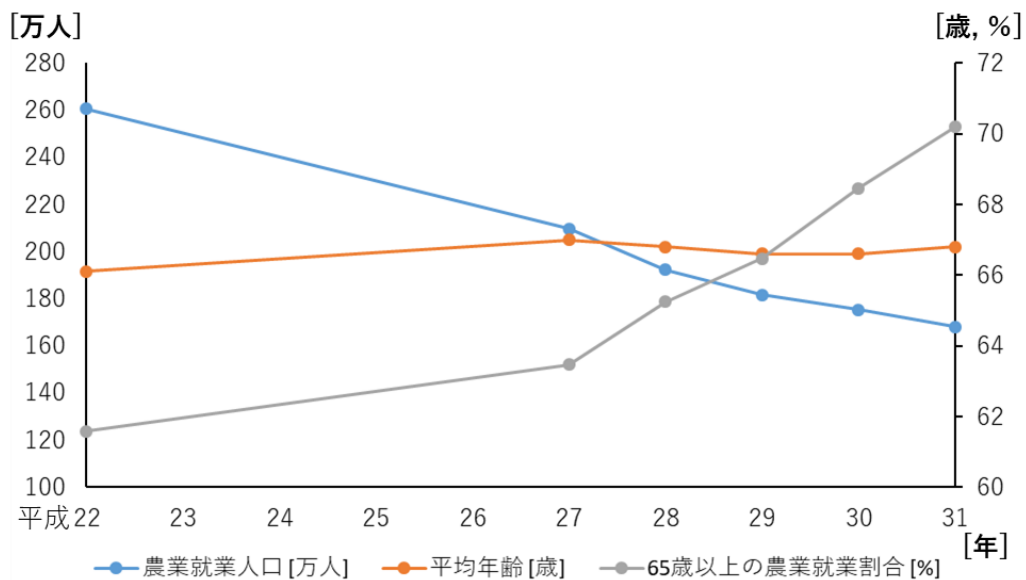


図 1.1：農業就業人口と平均年齢の推移

（農林水産省 HP よりデータを引用）

2. 解析データ

我々が解析を行うデータは、JA 全農ぐんま様（以下、JA 群馬様と表記する。）が保有する園芸作物生産実証農場（パイプハウス）の栽培環境となっており、複合環境制御装置（アグリネット+環境制御盤 MC-6000）によって 15 分おきにデータが計測されている。



（引用：JA 全農ぐんま HP^[1]）

図 2.1：園芸作物生産実証農場

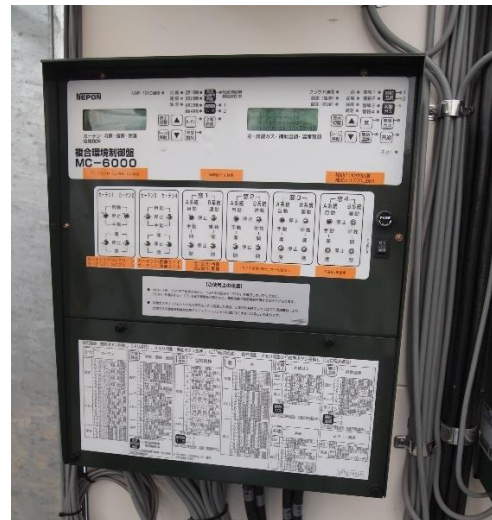


図 2.2：複合環境制御装置

2.1. データの種類

JA 群馬様が提供して下さるデータは、次の通りである。

- 気温 [°C]
- 炭酸ガス濃度 [ppm]
- 地中温度 [°C]
- 土壌水分 [%]
- 飽差 [g/m³]
- 絶対湿度 [g/m³]
- 積算日射量 [KJ/m²]

これら 7 変数は全て同時間に計測されており、データ数はそろっている。本稿ではこれらを基にして解析を行っていく。

3. 多変量解析^[2]

JA 群馬様よりご提供いただいているパラメータは7つある。これらから1つの収穫量を算出するには多変量を同時に解析できる技術を身に付ける必要がある。そこで本章では多変量を同時に解析する技術である、多変量解析について示していく。

3.1. 多変量解析の種類

多変量解析と一口に言っても、さまざまな手法が確立されている。この多変量解析を整理するにはまず、外的基準があるかないかで分類することができる。多変量解析にかけようとしているデータは複数の観測変数から構成されている。この複数の変数を原因となる変数と結果となる変数とに区別して分析する手法と、区別しないで分析する手法がある。区別する場合、原因側の変数を説明変数（独立変数・アイテム）などと呼び、結果側の変数を外的基準（被説明変数・目的変数・応答変数・従属変数・グループ化変数）と呼ぶ。以上にはさまざまな変数の呼び方を書き並べたが、多変量解析の手法によって呼び方が変わる。以下の表 3.1 に組み合わせを示す。

表 3.1：対応する変数の名称（外的基準のある多変量解析）

| 主な手法例 | 原因側の変数 | 結果側の変数 |
|----------------------|--------|-----------------|
| 数量化理論ⅠⅡ | アイテム | 外的基準 |
| 重回帰分析，判別分析 | 説明変数 | 被説明変数，目的変数，応答変数 |
| 重回帰分析 ロジスティック回帰分析 | 独立変数 | 従属変数 |
| 判別分析 | 独立変数 | グループ化変数 |

一方，因果関係を区別しない分析手法（外的基準が無い手法）では変数に名前がついていない。

3.2. 外的基準のある多変量解析

外的基準のある多変量解析にはさまざまな手法がある。その手法の一部を以下に示す。また、表 3.2 にはそれらの手法の説明変数と外的基準の測定尺度をまとめた。

(1) 重回帰分析

原因となる複数の説明変数が、結果となる 1 つの被説明変数（外的基準）に対してどのような影響を与えているのかを明らかにする分析方法。この分析では被説明変数、説明変数ともに量的データである必要がある。2 値のダミー変数（0/1）に変換すれば質的データでも利用可能である。

(2) 数量化 I 類（数量化理論第 I 類）

被説明変数が量的データで説明変数が質的データの場合に用いる簡易型の重回帰分析である。重回帰分析で質的データを用いる場合には 2 値変数に変換する必要があるが、この手法ではそのまま分析できる。また、日本の統計学者によって開発されたものであり、便利である反面、海外ではあまり認識されていない。

(3) ロジスティック回帰分析（ロジット分析）

被説明変数が質的データの重回帰分析である（説明変数は量的データ）。若干理論が異なるプロビット分析という手法もあるが、特に区別する必要は無い。単にロジスティック回帰分析という場合には被説明変数が 2 値の場合の分析手法を指すが、3 値以上のカテゴリデータの多項ロジスティック回帰分析や順位データの順序ロジスティック回帰分析（順序回帰分析）などさまざまな拡張版も存在する。

(4) 対数線型モデル

被説明変数だけでなく説明変数も質的データの場合の重回帰分析で、変数の内容に沿って様々な拡張版が存在する。

(5) 判別分析

いくつかの既知のグループ（外的基準）があるとき、ある標本がどのグループに属するかを推測する。つまり、外的基準が原因、説明変数が結果となる場合も多々ある。なお、外的基準は質的データを扱えるが、説明変数は量的データの場合がある。

(6) 数量化Ⅱ類

被説明変数，説明変数ともに質的データ版の判別分析である。対数線型モデルとほぼ同じ目的で用いることができるが，他の数量化理論と同様に海外では認知されていなく，説明変数の影響について検定ができないことから，対数線型モデルを使えるソフトがあるならばそちらを使ったほうが良い。

(7) 決定木（多段層別分析）

いくつかの説明変数（質的・量的を問わない）を階層的に組み合わせることで，被説明変数（2値変数）を分類する。標本を分類するルールを発見し，それを判別や予測に生かすことが目的である。判別分析同様，変数の因果関係が逆転していても問題ない。

表 3.2：外的基準のある多変量解析の測定尺度

| 手法 | 外的基準（被説明変数） | 説明変数 |
|-------------|-------------|-------|
| 重回帰分析 | 量的データ | 量的データ |
| 数量化Ⅰ類 | 量的データ | 質的データ |
| ロジスティック回帰分析 | 質的データ | 量的データ |
| 対数線型モデル | 質的データ | 質的データ |
| 判別分析 | 質的データ | 量的データ |
| 数量化Ⅱ類 | 質的データ | 質的データ |
| 決定木 | 質的データ | 質的データ |

3.3. 外的基準のない多変量解析

外的基準のない多変量解析の目的は，変数があまりに多く処理が複雑になってしまう場合に，それらを2～3の新しい変数にまとめ，その新しい変数を使って標本を分類したり解釈したりすることである。たいていは標本や変数を平面図などにマッピングして視覚的に捕らえやすくする。

外的基準のない多変量解析には量的データを対象とした手法として主成分分析，因子分析，クラスター分析などがある。一方，質的データも扱える手法としてコレスポンデンス分析（数量化理論Ⅲ類）や多次元尺度法（数量化理論Ⅳ類）などがある。

(1) 主成分分析

たくさんの変数（量的データ）から、標本の性質をよく説明するような少数の変数を合成する。標本ごとに得点を求めることができるため、他の多変量解析への2次利用も容易である。

(2) 因子分析

潜在的な共通因子を発見する。主成分分析と理論は異なるものの同じ目的で使われることが多い。

(3) クラスタ分析

標本を似ている同士でまとめて行き、いくつかのグループに分類する。標本だけでなく変数の分類にも使われる。理論的には質的データを使用することが可能だが、対応するソフトが少ないため一般的には量的データを用いる。

(4) コレスポネンシ分析

質的データを簡単に扱えるようにした主成分分析である。ピアソンの χ^2 検定のように集計済みのクロス集計表からでも分析ができる。ほぼ同じ内容で数量化Ⅲ類があるが、これは海外であり認知されていない。

(5) 多次元尺度（構成）法

質的変数から標本に何らかの類似度を定義して、平面図にマッピングする。類似度の測定尺度によっていろいろな手法があり、数量化Ⅳ類もその1つといえる。数学的に洗練されているため恣意性が入りにくいという長所をもつが、（計算が複雑であるため）搭載されているソフトは少ない。

表 3.3：外的基準のない多変量解析における測定尺度

| 手法 | 外的基準（被説明変数） | 説明変数 |
|-----------|-------------|-------|
| 主成分分析 | なし | 量的データ |
| 因子分析 | なし | 量的データ |
| クラスタ分析 | なし | 量的データ |
| コレスポネンシ分析 | なし | 質的データ |
| 多次元尺度法 | なし | 質的データ |

3.4. その他の多変量解析

以上に挙げた手法のほかに、正準相関分析や共分散構造分析がある。

(1) 正準相関分析

2組の重回帰モデルを設定し、その2つの被説明変数同士の相関係数が最大になるように変数の重みを推定する手法である。2つのグループに分けられた変数同士の関連性を解明できる。

(2) 共分散構造分析（構造方程式モデリング，SEM）

複数の因子分析を1つのモデル内で同時に推定する手法である。結果をパス図で表せるので、図中の観測変数や潜在変数の位置から問題の構造や間接的な影響力の強さなどを容易に探ることができる。グラフィカルモデリングという探索手法を併用することで分析者によるモデル構造の恣意性を改善できる。

3.5. 多変量解析の手法選択フローチャート

多変量解析の種類をここまで紹介してきたが、これだけの種類があつてはどの手法を用いてよいかわからない。そこで、多変量解析の手法を選ぶためのフローチャートとして図3.1を示す。

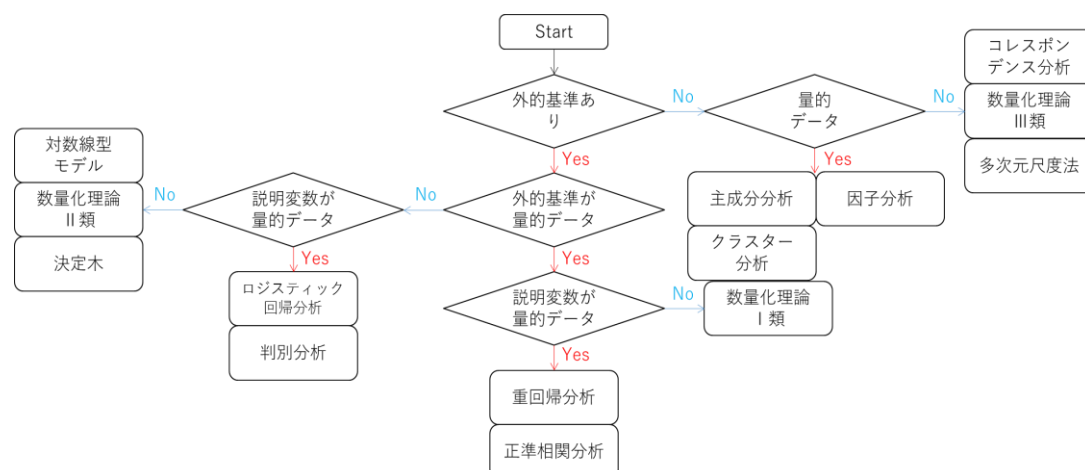


図 3.1：多変量解析の選択フローチャート

本稿の研究では、説明変数（環境パラメータ）と外的基準（収穫量）に明らかに分かれており、環境パラメータおよび収穫量がともに量的データであるので、図 3.1 から重回帰分析もしくは正準相関分析が適切であることがわかる。本研究ではこの中から重回帰分析を選択した。

4. 重回帰分析

重回帰分析とは前章で述べたように、原因となる複数の説明変数が結果となる1つの被説明変数（外的基準）に対してどのような影響を与えているのかを明らかにする分析方法であり、一般的な形として次の(4.1)式のような形で表せる。

$$y = a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n + b \quad (4.1)$$

この式において y は目的変数（結果側）、 x は説明変数（原因側）、 a は偏回帰係数、 b は誤差である。この x や y に複数のデータセットを入力することで、 a や b の最適値を算出していく。得られた a, b を用いることで、ある環境下で農作物を育成した時の収穫量を予測できるようになる。

本稿においては、 y をきゅうりの収穫量[kg]、 x を環境パラメータとした。

4.1. 目的変数 y

目的変数 y として、JA 群馬様よりいただいたきゅうりの出荷実績の収穫量を扱った。このデータを詳しく見ていくと、各月に数回、収穫が全く行われていない日が数日あった。これはJA 群馬様の休日であることが考えられるが、この日の収穫量を単純に0と考えると、きゅうりの収穫傾向に大きな誤差を生んでしまうことが予想される。また休日の翌日においても、前日に収穫できるはずであったきゅうりを持ち越して収穫しているため、他の日に比べて収穫量が明らかに大きい。

そこで、JA 群馬様が収穫を一切行っていない日とその翌日を除いたデータの散布図を作成し、その近似曲線を算出した。この近似曲線を用いることで、収穫を行っていない日とその翌日を補完し、収穫量とした。

4.2. 説明変数 x

重回帰分析において、偏回帰係数 a を求めるためには説明変数1組に対して、目的変数が1つの1:1対応でなければ計算ができない。しかし、説明変数（環境データ）は15分につき1データ、目的変数（収穫量データ）は1日につき1データとなっている。さらに農作物の収穫量に影響を与える環境データは収穫日には依存せず、芽がでて花びらをつけ、実になり、収穫できる大きさになるまでの数日分も考慮していかなければならない。よって何らかの処理を行うことで、データ数を1:1対応にしていく必要がある。それを本稿では、積分法を用いることで解決する。具体的には次のようになる。

きゅうりの収穫時期は、開花から12日とされている。そこで、きゅうりを収穫した日を0日とし、推定開花日である-12日までの各変数（気温・炭酸ガス等）の積分値（24時間×12日分の総和）を説明変数に与えた。なお、積分計算には台形則を用いた。（図4.2）

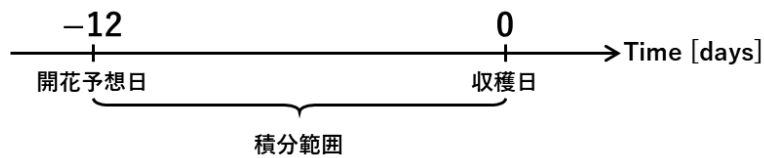


図 4.1：積分範囲

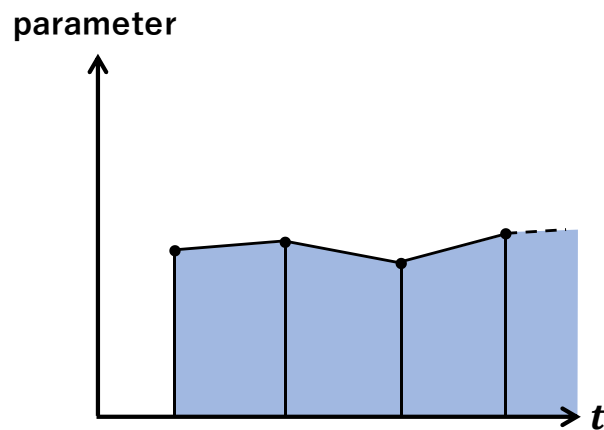
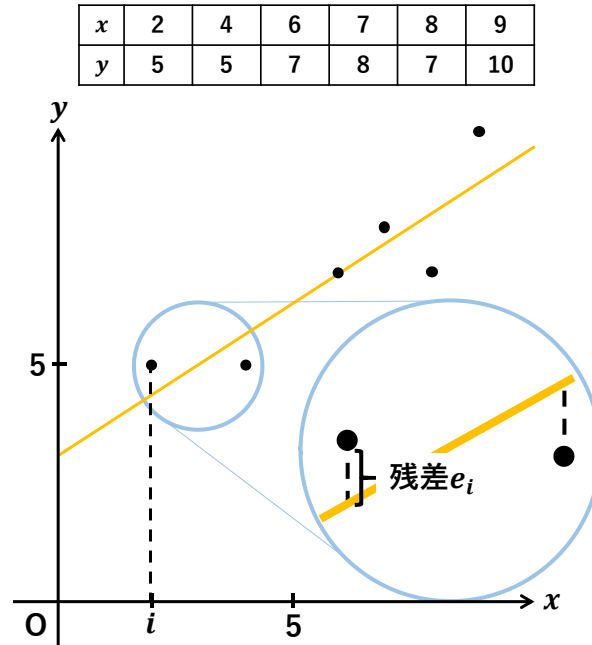


図 4.2：台形則

この台形則を用いることによってパラメータごとに積算値を求め、これを説明変数側の1データとした。

4.3. 回帰式の評価方法^[3]

重回帰分析における評価として決定係数と呼ばれる指標を用いる。



(この図で用いられる値は説明の為の架空な値である)

図 4.3 : 評価方法 (決定係数)

図 4.3 中の回帰直線 (黄色線) を求めると $y = 0.647x + 3.118$ であった。 (x_i, y_i) に対する予測値を \hat{y}_i , 予測値 \hat{y}_i と実測値 y_i との差を残差 e_i , 実測値 y_i の平均値を \bar{y} と書くことにすれば, 決定係数 R^2 は次のように表すことができる。

$$R^2 = 1 - \frac{\sum_{i=1}^n e_i^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{S_e}{S_{yy}} \quad (4.2)$$

(S_e : 残差の平方和, S_{yy} : y の偏差の平方和)

この決定係数 R^2 は 0 から 1 までの値をとり, 1 に近いほど回帰直線による予測が良く当たるといことになる。

以上に示す決定係数は回帰直線の精度を示すことで間違いないが、重回帰分析を考える場合、決定係数が大きくてもそれが必ずしも回帰直線の精度が高いと示せない。これは説明変数を増やすとその説明変数が妥当であるか否かにかかわらず、決定係数が増加する傾向にあるからである。この問題点を解決するために、「自由度修正済み決定係数 R_f^2 」と呼ばれる指標が存在する。自由度修正済み決定係数は次の(4.3)式で示される。

$$R_f^2 = 1 - \frac{\frac{\sum_{i=1}^n e_i^2}{n-p-1}}{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}} = 1 - \frac{\frac{S_e}{n-p-1}}{\frac{S_{yy}}{n-1}} \quad (4.3)$$

(S_e : 残差の平方和, S_{yy} : y の偏差の平方和, n : データ数, p : 説明変数の数)

これら2つの評価指標により、回帰式の妥当性を評価する。

4.4. 説明変数パラメータの改良

JA 群馬様より頂いた環境データ7変数をそのまま用いたのではあまり良い結果が得られないと予想される。というのもJA 群馬担い手サポートセンターの方々とのミーティングで、きゅうりの育成には①一日の気温差が重要、②日出時と日没時で育成方法を変えるべきなど、様々な現場の栽培方法を教えていただいていた。このことから、解析する説明変数には時系列情報（日中と日没）を含めていく必要がある。そこで7変数に時系列情報を付加する処理を行うことにした。その内容を本節で示す。

4.4.1. 日出時刻と日没時刻を用いたパラメータの2分割

日中における気温と日没後における気温，光合成を行える日中における炭酸ガス濃度と呼吸のみを行う日没後における炭酸ガス濃度のように，気温と炭酸ガス濃度パラメータは太陽の有無でパラメータを分けたほうが良いと考えられる。そこで，日出・日没時刻を用いることによってパラメータを2分割することにした（図4.4）。このようにすることで，図4.5のように日中と日没それぞれで積分ができるので，特徴量が見えやすくなる。

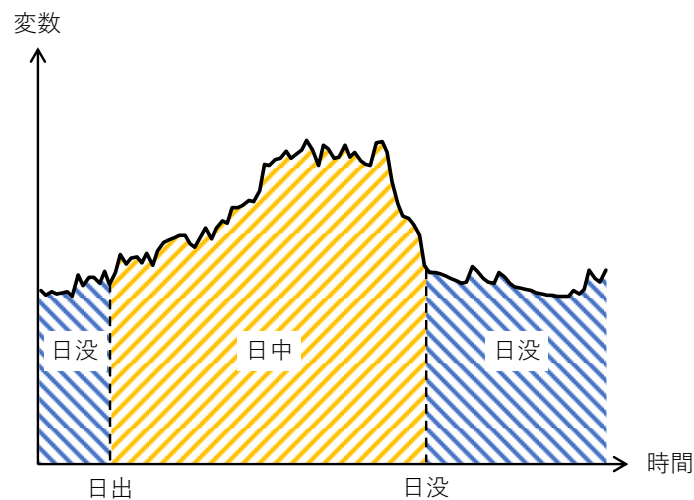


図 4.4 : 日出と日没に 2 分割

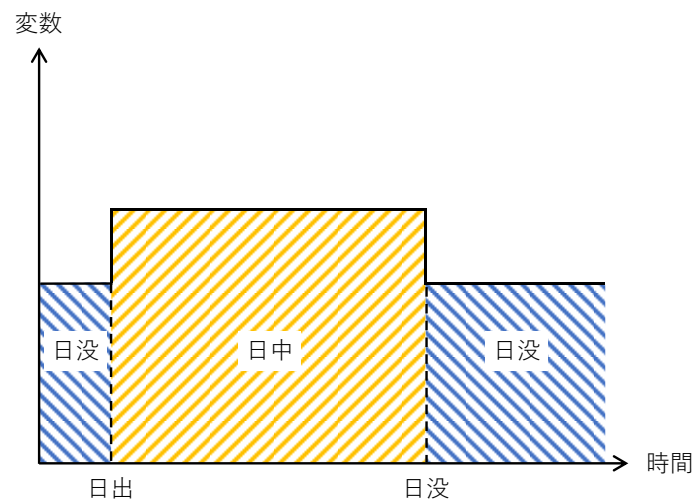


図 4.5 : 積分後の見え方

4.4.2.一日の気温差

図 4.4, 図 4.5 に示すように日中と日没に分けて積分を行っても日中, 日没それぞれで平坦になってしまい, 一日の気温差がわからなくなってしまう。そこで最高気温と最低気温の差を算出し, 「気温差」という新たなパラメータを設けた。

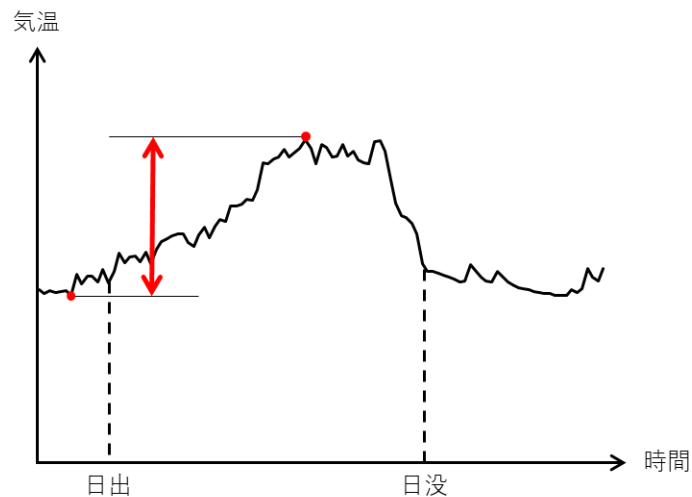


図 4.6 : 最高気温と最低気温の差

4.4.3.適正温度時間長

きゅうりを栽培するのに適した気温が存在する。それが, 以下の表 4.1 である。

表 4.1 : きゅうり栽培の好条件

| | | |
|------|-----------|-------------|
| 発芽適温 | 25 ～ 30 度 | |
| 育成適温 | 昼間 | : 22 ～ 28 度 |
| | 夜間 | : 17 ～ 18 度 |

表 4.1 に示した育成適温にハウス内を設定できていれば, きゅうりはよく育つと考えられる。そこで, 昼間の育成適温に合致する時間長, 夜間の育成適温に合致する時間長の積算値を 1 つのパラメータにした。

これにより, 説明変数として扱うパラメータは次の通りとなった。(表 4.2)

表 4.2：重回帰分析に用いる変数

| 使用するパラメータ | | |
|-----------|----------|------|
| 土壌温度 | 土壌水分 | 飽差 |
| 絶対湿度 | 日中気温 | 夜間気温 |
| 日中炭酸ガス濃度 | 夜間炭酸ガス濃度 | 気温差 |
| 日射量 | 適正温度時間長 | |

4.5. 多重共線性の考慮

表 4.2 に示した変数を重回帰分析で用いていくことにした。これらを重回帰分析における説明変数として扱う際に、「多重共線性」と呼ばれる現象が起きてしまう可能性があるため、先に確認をしていく。

4.5.1. 多重共線性とは^{[4],[5]}

重回帰分析やロジスティックス回帰分析などの多変量解析を行ったときに、互いに関連性の高い説明変数が存在すると、解析上の計算が不安定になり回帰式の精度が極端に悪くなってしまうたり、回帰係数などが異常な値をとったりする場合がある。このように解析結果が不安定な状態となる現象を多重共線性（Multicollinearity, マルチコ）と呼ぶ。

多重共線性が起こりやすい条件として、

- ① 説明変数間の相関係数が ± 1 に近い組み合わせが含まれている。
- ② 説明変数の個数がサンプルサイズに比べて大きい。

などが考えられる。

4.5.2. 多重共線性の確認

多重共線性が生じていないかを確認する方法にはさまざまな方法がある。その中でも容易に確認できる方法として、①相関係数や②VIF (Variance Inflation Factor) が存在する。本稿ではこれら 2 点を用いることにより、多重共線性の評価を行う。

① 相関係数

相関係数 r とは、2 つの変数同士がどのような関係性を持っているのかを確認するための評価方法であり、次の(4.4)式で表される。

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.4)$$

この式により相関係数 r は -1 から 1 の範囲で表され、一般的に $0.7 \leq r \leq 1$ を「正の相関関係がある」、 $-1 \leq r \leq -0.7$ を「負の相関関係がある」と呼び、多重共線性を引き起こしている可能性があると考えられる。

実際に相関係数を求めることで多重共線性の評価を行っていく。

相関係数に用いる環境パラメータは、表 4.2 に示した変数を 4.2 節で示した積分法で処理したものを用いた。なお、積分の範囲はシンプルなものとするため、1 日とした。この結果を表 4.3 に示す。 $(0.7 \leq |r| \leq 1)$ の範囲は赤字でマークした。

表 4.3 を見ると、「土壌温度－絶対湿度」「土壌温度－日中気温」「絶対湿度－日中気温」「日中気温－夜間炭酸ガス濃度」の 4 箇所が赤字でマークされていることから、多重共線性を引き起こしていると考えられる。

表 4.3 : 環境パラメータ間の相関性

| | 土壌温度 | 土壌水分 | 飽差 | 絶対湿度 | 日中気温 | 夜間気温 | 日中炭酸ガス濃度 | 夜間炭酸ガス濃度 | 気温差 | 日射量 | 適正温度時間長 |
|----------|--------|--------|--------|--------|--------|--------|----------|----------|--------|--------|---------|
| 土壌温度 | | 0.347 | 0.162 | 0.852 | 0.866 | 0.396 | -0.143 | -0.544 | -0.166 | 0.386 | -0.296 |
| 土壌水分 | 0.347 | | 0.083 | 0.190 | 0.213 | 0.339 | -0.220 | -0.404 | -0.042 | 0.052 | 0.100 |
| 飽差 | 0.162 | 0.083 | | -0.231 | 0.202 | 0.226 | -0.025 | -0.014 | 0.491 | 0.385 | -0.402 |
| 絶対湿度 | 0.852 | 0.190 | -0.231 | | 0.850 | 0.148 | -0.120 | -0.560 | -0.292 | 0.252 | -0.323 |
| 日中気温 | 0.866 | 0.213 | 0.202 | 0.850 | | -0.029 | -0.205 | -0.702 | -0.048 | 0.423 | -0.462 |
| 夜間気温 | 0.396 | 0.339 | 0.226 | 0.148 | -0.029 | | 0.176 | 0.267 | -0.145 | 0.053 | 0.013 |
| 日中炭酸ガス濃度 | -0.143 | -0.220 | -0.025 | -0.120 | -0.205 | 0.176 | | 0.675 | -0.148 | -0.364 | 0.085 |
| 夜間炭酸ガス濃度 | -0.544 | -0.404 | -0.014 | -0.560 | -0.702 | 0.267 | 0.675 | | 0.080 | -0.254 | 0.217 |
| 気温差 | -0.166 | -0.042 | 0.491 | -0.292 | -0.048 | -0.145 | -0.148 | 0.080 | | 0.419 | -0.272 |
| 日射量 | 0.386 | 0.052 | 0.385 | 0.252 | 0.423 | 0.053 | -0.364 | -0.254 | 0.419 | | -0.519 |
| 適正温度時間長 | -0.296 | 0.100 | -0.402 | -0.323 | -0.462 | 0.013 | 0.085 | 0.217 | -0.272 | -0.519 | |

② VIF

VIFとは説明変数間の多重共線性を検出するための指標の1つである。説明変数間の相関係数行列の逆行列の体格要素であり、値が大きい場合はその変数を分析からのぞいたほうが良いと考えられる。また、VIFには説明変数間の相関係数を用いた算出方法と重回帰分析で求めた決定係数による算出方法の2通りがある。前者の算出には表4.3に示した相関係数 R^2 を以下に示す(4.5)式に代入するだけでよい。そのため、わざわざ式へ代入せずに相関係数 R^2 で評価すればよいことになる。さらに前者はパラメータ2つの相関性を見ているだけで、複数のパラメータ間での相関性は見られない。そこでVIFでの評価は後者のみを行うことにする。

$$VIF = \frac{1}{1 - R^2} \quad (4.5)$$

重回帰分析で求めた決定係数による算出方法について説明する。重回帰分析は前節で示したように、(4.1)式のように表せる。VIFでは環境パラメータ内のいずれか1つを y へ代入し、それ以外の環境パラメータで回帰式を求める。その際の決定係数 R^2 （あるいは自由度修正済み決定係数 R_f^2 ）を(4.5)式へ代入することでVIF値を算出していく。このようにして算出されたVIF値を以下の表に示す。

表 4.4 : VIF 結果

| | 決定係数 R^2 | VIF |
|----------|------------|--------|
| 土壌温度 | 0.954 | 21.798 |
| 土壌水分 | 0.602 | 2.514 |
| 飽差 | 0.952 | 20.664 |
| 絶対湿度 | 0.984 | 60.659 |
| 日中気温 | 0.988 | 80.790 |
| 夜間気温 | 0.935 | 15.482 |
| 日中炭酸ガス濃度 | 0.774 | 4.427 |
| 夜間炭酸ガス濃度 | 0.912 | 11.361 |
| 気温差 | 0.536 | 2.155 |
| 日射量 | 0.608 | 2.548 |
| 適正温度時間長 | 0.625 | 2.669 |

表 4.4 は左列に示した環境パラメータを(4.1)式の y 、それ以外のパラメータを x としたときの決定係数及びVIFである。統計学的にはVIF値が10より大きくなると多重共線性が生じている可能性が高いと判断される。表を見ると、多くのパラメータでVIF値が10を上

回っており，これでは重回帰分析の結果が不安定になり得る。そこで相関係数と VIF の結果を踏まえ，パラメータの厳選を行っていく。

①の相関係数では，「土壌温度」「絶対湿度」「日中気温」の3変数は互いに高い相関性を持っており，②の VIF においてもこれら3変数は大きな値を持つため，「絶対湿度」のみを残し，他の2変数はカットした。残されたパラメータのみで再度 VIF を算出した結果を以下に示す。

表 4.5 : VIF 結果 (2 変数カット)

| | 決定係数 R^2 | VIF |
|----------|------------|--------|
| 土壌水分 | 0.562 | 2.284 |
| 飽差 | 0.692 | 3.248 |
| 絶対湿度 | 0.813 | 5.357 |
| 夜間気温 | 0.721 | 3.591 |
| 日中炭酸ガス濃度 | 0.759 | 4.146 |
| 夜間炭酸ガス濃度 | 0.902 | 10.247 |
| 気温差 | 0.497 | 1.987 |
| 日射量 | 0.544 | 2.195 |
| 適正温度時間長 | 0.474 | 1.901 |

表 4.4 では6つのパラメータで VIF 値が10を上回っていたが，2変数のカットすることで，1つまで減らすことができた。最後に，VIF 値が10を上回っている夜間炭酸ガス濃度をカットすることで，多重共線性の可能性を消すことができた。最終結果を表 4.6 に示す。

表 4.6 : VIF 結果 (3 変数カット)

| | 決定係数 R^2 | VIF |
|----------|------------|-------|
| 土壌水分 | 0.255 | 1.342 |
| 飽差 | 0.503 | 2.011 |
| 絶対湿度 | 0.458 | 1.846 |
| 夜間気温 | 0.305 | 1.438 |
| 日中炭酸ガス濃度 | 0.263 | 1.357 |
| 気温差 | 0.433 | 1.762 |
| 日射量 | 0.501 | 2.005 |
| 適正温度時間長 | 0.473 | 1.897 |

以上に示した過程で、3つのパラメータをカットした。これにより重回帰分析の説明変数として扱うパラメータが決定した。その全てを表4.7に示す。

表 4.7：厳選された変数（8 変数）

| 使用するパラメータ | | |
|-----------|----------|------|
| 土壌水分 | 飽差 | 絶対湿度 |
| 夜間気温 | 日中炭酸ガス濃度 | 気温差 |
| 日射量 | 適正温度時間長 | |

4.6. 日射量の考慮

JA 群馬様の試験農場は表 4.7 に示した変数のうち日射量のみが制御できないため、日射量が少ない時にどのような栽培を行っていけばよいか確認してほしいと要望を受けた。しかし、日射量が「少ない」という表現は抽象的すぎるため、統計的に「多い」「少ない」を定義した。

日射量の「多い」「少ない」を統計的に定義するために、正規分布（ガウス分布）と呼ばれる指標を取り入れる。正規分布とは、図 4.7 のような左右対称の連続型の確率分布である。正規分布は最も代表的な分布の 1 つであり、例えば物理などの実験における測定誤差、テストの点数などは正規分布に従うことが多いと考えられている。

正規分布の確率密度関数は式(4.6)のように表される。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.6)$$

μ ：平均値， σ^2 ：分散（ σ は標準偏差）

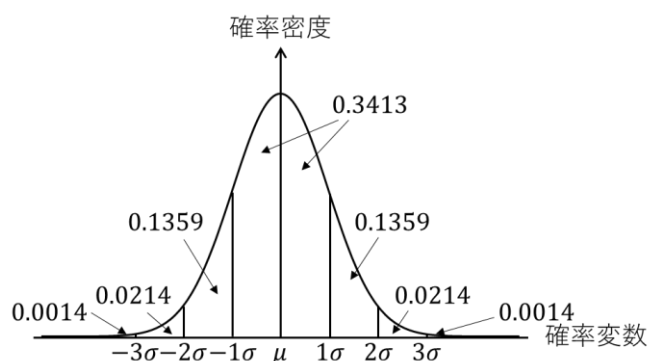


図 4.7：正規分布の面積（確率）と標準偏差

また正規分布において、 $[-\sigma, \sigma]$ を 1σ 区間と呼び、約 68%のデータがこの範囲に該当する。同様に $[-k\sigma, k\sigma]$ を $k\sigma$ 区間と呼び、 2σ 区間に入る確率は約 95%、 3σ 区間に入る確率は約 99.7%となる。

日射量が正規分布に従うものであるかを確認してみたところ、図 4.8 のような結果となった。図 4.8 を見ればわかるが、理想に近い正規分布をしていることが分かった。そこで、上述したような σ を用いて日射量を分類した。定義を表 4.8 および図 4.9 に示す。

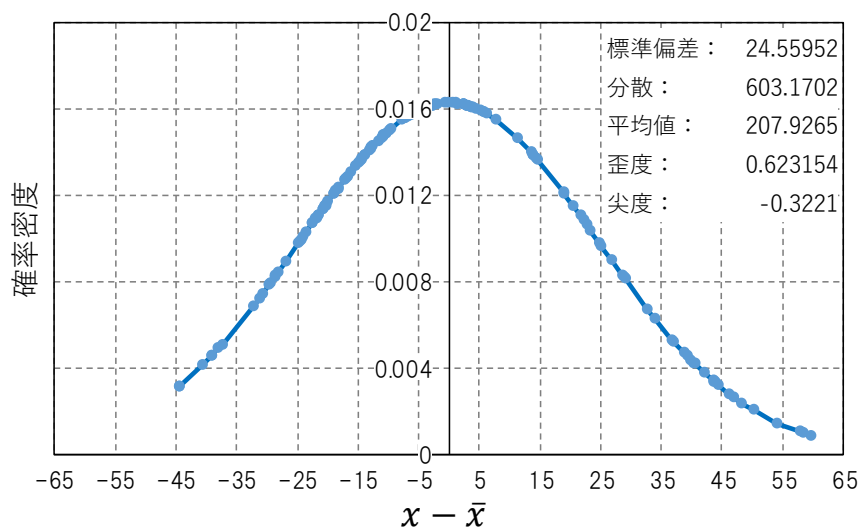


図 4.8：日射量データ

表 4.8：日射量が「多い期間」と「少ない期間」の定義

| 日射量 | 定義 | 割合 |
|-----|-----------------------------------|------|
| 多い | $x \geq \mu + \sigma$ | 16 % |
| 普通 | $\mu - \sigma < x < \mu + \sigma$ | 68 % |
| 少ない | $x \leq \mu - \sigma$ | 16 % |

※ x は日射量を表す。



図 4.9：日射量の定義

4.7. 解析結果

前節で厳選してきたパラメータを用いて、重回帰分析を行っていく。重回帰分析には Python 言語のライブラリである、`scikit-learn` を用いた。

解析条件を以下に示す。

表 4.9：解析条件

| | |
|---------|--|
| 解析期間 | 2017 年促成栽培データ：2016/12/26 ～ 2017/07/10 2018 年促成栽培データ：2018/01/16 ～ 2018/07/22 |
| 積分範囲 | 12 日間 |
| 使用パラメータ | (表 4.7 参照) |

この条件下で解析を行った結果が以下に示す表 4.10 である。

表 4.10：解析結果

| 日射量による分類 | - | 多 | 普 | 少 |
|-------------------------|-------------|-------------|-------------|-------------|
| 土壌水分 hum_soil | 8.807.E-06 | 2.016.E-05 | 8.890.E-06 | 1.382.E-05 |
| 飽差 VPD | 8.513.E-05 | 2.596.E-05 | 9.209.E-05 | 2.241.E-05 |
| 絶対湿度 amount_of_water | 3.803.E-05 | 2.822.E-06 | 4.120.E-05 | -3.059.E-06 |
| 夜間気温 nighttime_tem | -2.966.E-05 | 7.841.E-05 | -3.906.E-05 | 1.107.E-04 |
| 日中炭酸ガス濃度 daytime_CO2 | -3.165.E-06 | -2.149.E-06 | -3.002.E-06 | -9.790.E-08 |
| 気温差 tem_dif | 2.175.E-01 | 4.961.E-01 | 2.604.E-01 | -5.408.E+00 |
| 日射量 solar | -2.789.E-01 | 6.608.E-01 | -3.305.E-01 | 2.004.E+00 |
| 適正温度時間長 suitable_tem | 8.203.E-04 | 7.236.E-04 | 8.378.E-04 | 4.060.E-04 |
| 誤差 | -155.432 | -1083.490 | -203.051 | -938.800 |
| R^2 | 0.663 | 0.892 | 0.638 | 0.675 |
| R_f^2 | 0.654 | 0.874 | 0.622 | 0.589 |
| データ数 | 293 | 56 | 198 | 39 |

日射量に関して正規分布によりデータを3分割した場合の重回帰分析（表 4.10 右3列）をみると、「飽差」「日中炭酸ガス濃度」「適正温度時間長」の3変数は、全ての場合において偏回帰係数の符号が一致している。しかし、「夜間気温」や「日射量」は偏回帰係数の符号が一致しておらず、矛盾が生じている。「夜間気温」について見れば、日射量が多いときと少ないときは夜間気温を上げたほうが好ましく、日射量が普通となるときは下げたほうが良いということになる。これは「日射量」についても同じことが言える。このような結果が生じてしまった原因には、データを日射量に関して3分割した際に夜間気温や日射量データのレンジ（範囲）が狭まり、その結果データの相関性が高まってしまった、すなわち再び多重共線性を生じてしまったことが考えられる。そこで、次節に日射量を3分割した際の多重共線性の有無を検証していく。

4.8. 多重共線性の再検証

表 4.10 において生じた矛盾の原因を確認するために、再度多重共線性の再検証を行っていく。4.5.2 項と同様の方法で日射量が多い時、少ない時を分けた状態で検証した。この時の結果が表 4.11 である。

表 4.11：多重共線性の再検証

| 日射量による分類 | 多 | 普 | 少 |
|-------------------------|--------|--------|--------|
| 土壌水分 hum_soil | 30.035 | 2.926 | 4.577 |
| 飽差 VPD | 19.278 | 3.190 | 4.958 |
| 絶対湿度 amount_of_water | 18.494 | 12.119 | 12.638 |
| 夜間気温 nighttime_tem | 16.013 | 1.974 | 9.761 |
| 日中炭酸ガス濃度 daytime_CO2 | 17.967 | 2.052 | 6.425 |
| 気温差 tem_dif | 11.165 | 3.927 | 6.916 |
| 日射量 solar | 2.918 | 1.749 | 2.053 |
| 適正温度時間長 suitable_tem | 28.481 | 7.372 | 9.053 |

表 4.11 を見ると VIF が 10 を上回る部分が多く、予想通りの結果であることがわかる。そのため、日射量別にパラメータを改めて厳選し重回帰分析を行っていく必要がある。しかし、JA 群馬様より頂いている要望は「日射量が少ないときに環境条件をどのようにするのが好ましいのか」という点であった為、日射量が「多い」部分は本稿において無視し日射量が「少ない」場合の説明変数を再度厳選しなおすこととした。

4.9. 日射量が「少ない」時の重回帰分析

表 4.11 を見ると、日射量が「少ない」時の VIF では絶対湿度が 10 を上回っていることが確認できる。そこで、絶対湿度パラメータをカットした 7 変数で重回帰分析を再度行ってみた。その結果を表 4.12 に示す。

表 4.12：日射量が「少ない」時の VIF 及び重回帰分析

| | VIF | 重回帰分析 |
|-------------------------|-------|-------------|
| 土壌水分 hum_soil | 4.561 | 1.377.E-05 |
| 飽差 VPD | 3.686 | 2.451.E-05 |
| 夜間気温 nighttime_tem | 9.282 | 1.084.E-04 |
| 日中炭酸ガス濃度 daytime_CO2 | 6.262 | -1.410.E-07 |
| 気温差 tem_dif | 3.493 | -5.197.E+00 |
| 日射量 solar | 2.050 | 2.012.E+00 |
| 適正温度時間長 suitable_tem | 3.313 | 4.544.E-04 |
| 誤差 | | -1021.310 |
| R^2 | | 0.675 |
| R_f^2 | | 0.602 |
| データ数 | | 39 |

絶対湿度をカットし VIF を求めることで、全ての変数が 10 より小さくなった。これにより、多重共線性の可能性を除去することができた。そこで、これら 7 変数を用いて重回帰分析を行ったところ、表 4.12 右列のような結果を得た。各パラメータの偏回帰係数を見ればわかるかと思うが、日射量が少ない時は土壌水分、飽差、夜間気温、適正温度時間長は積極的に増やすあるいは上げるようにし、日中炭酸ガス濃度、気温差は下げていくあるいは小さくしていくことが望ましいということが言える。これらのことは、大きく矛盾するような点が見られない為、妥当な結果が得られているのではないかと考えている。

しかし決定係数 R^2 および自由度修正済み決定係数 R_f^2 をみると、いずれも 0.6 ほどであった。

5. 収穫量予測

JA 群馬様より頂いた要望として、「数日後の収穫量予測をしたい」という点が上がった。この要望にお応えするために次のような方法を考案した。

5.1. 収穫量予測の手法

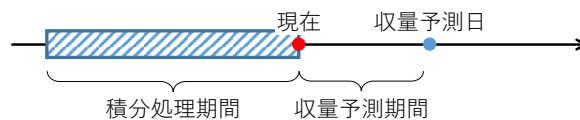
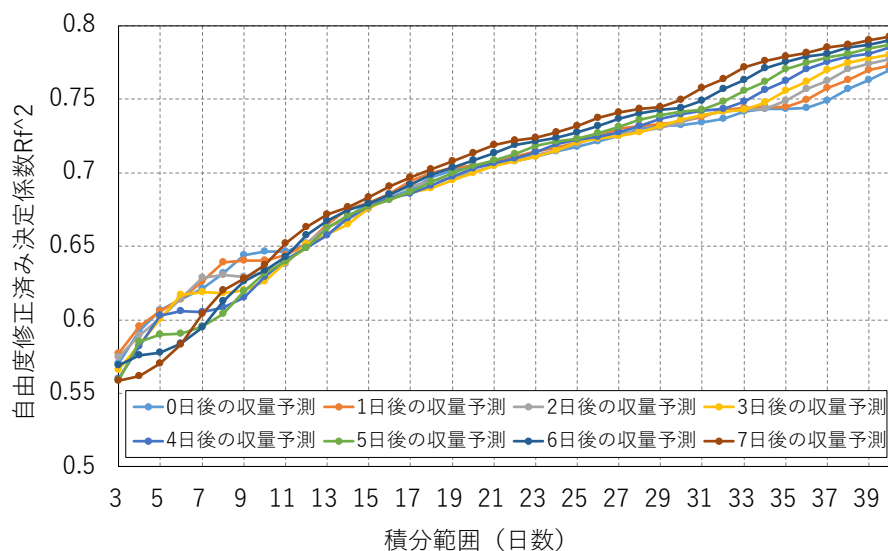


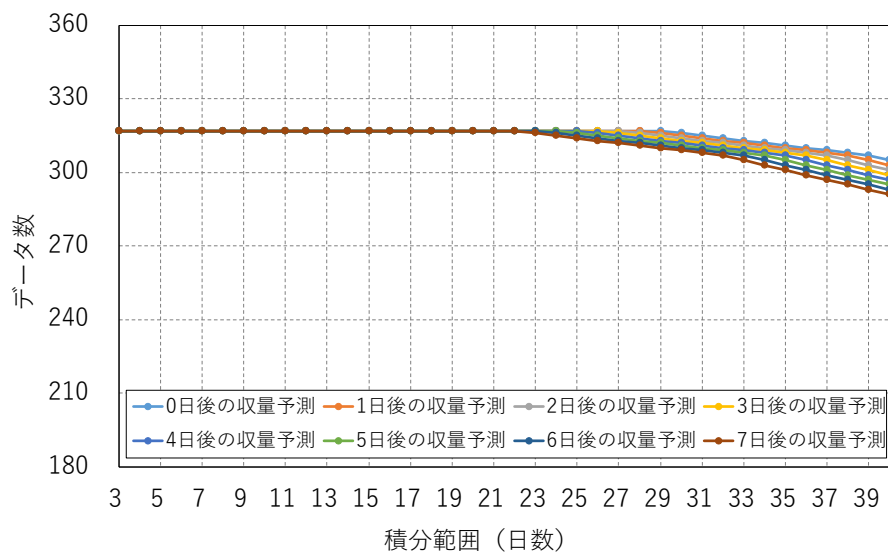
図 5.1：収穫量の予測手法

前章で重回帰分析を行ってきたように各環境パラメータの処理には積分を施し、その積分値から、数日後の収量を予測していく。しかし、収量を予測するための積分処理期間は何日間が妥当であるのか、その積分処理期間に対して何日後の予測が可能となるのかといった部分は見当がつかない。そこで「積分処理期間」と「収量予測期間」のパラメータを変更しながら、それぞれの決定係数値（収量予測値の信頼性）を算出してみた。その結果を図 5.2、図 5.3 に示す。



（2017-2018 年促成栽培データを使用）

図 5.2：収量予測における決定係数の推移



(2017-2018 年促成栽培データを使用)

図 5.3 : 収量予測におけるデータ数の推移

図 5.2 において、いずれの収量予測条件についても波形が右肩上がりとなっており、積分範囲(日数)は長ければ長いほど良いことがわかる。さらに積分範囲が 40 日の時を見れば、0 日後の収穫量予測より、7 日後の収穫量予測の方が高精度でできていることがわかる。重回帰分析では積分範囲が 12 日の時が最も良い結果であったが、収穫量予測という点では「積分範囲が伸びることで決定係数の値も伸びる」ことがわかった。

図 5.3 には、収量予測をする際に用いられるデータ数の推移を示した。

5.2. 収穫量予測の評価

前節の図 5.2 では「7 日後の収穫量を 40 日分の積分処理の結果で予測する」場合の回帰式が最も信頼性を得られるものだった。そこで、本節は「積分処理期間を 40 日」と「収量予測期間を 7 日」として、予測の評価を行っていく。表 5.1 には「積分処理期間 40 日、収穫量予測期間 7 日」の際の偏回帰係数と決定係数を示した。

表 5.1：収穫量予測における偏回帰係数
(積分処理期間 40 日，収量予測期間 7 日)

| | |
|-------------------------|-------------|
| 土壌水分 hum_soil | 5.583.E-06 |
| 飽差 VPD | 3.380.E-05 |
| 絶対湿度 amount_of_water | 1.296.E-05 |
| 夜間気温 nighttime_tem | -5.833.E-06 |
| 日中炭酸ガス濃度 daytime_CO2 | 2.035.E-07 |
| 気温差 tem_dif | 2.427.E-01 |
| 日射量 solar | -3.132.E-01 |
| 適正温度時間長 suitable_tem | 4.203.E-04 |
| 誤差 error | -2083.263 |
| 決定係数 R^2 | 0.798 |
| 自由度修正済み 決定係数 R_f^2 | 0.792 |
| データ数 number_of_data | 291 |

さて、収量予測を行うための回帰式を評価する値として「決定係数（自由度修正済み決定係数）」というものが存在する。だが、この指標ひとつだけを見てもどれだけ実際の収穫量と予測値に誤差が生じているのか定かではない。そこで、この回帰式が算出する収穫量（予測値）と実際の収穫量（実測値）を比較した。その結果が以下に示す図 5.4，5.5 となる。

図 5.4 の棒グラフは実測値を収穫量 0～600[kg]をレンジ 50[kg]で分類しデータ数をまとめたもの、折れ線グラフは予測値を実測値と同様にしておいたものである。実測値と予測値のデータ数には差が見られ、この回帰式には問題があるように感じられる。しかしこのグラフには欠点があり、例えば実際の収穫量が 348[kg]で予測値が 351[kg]だった場合、棒グラフでは 300～350[kg]にカウントされるのに折れ線グラフでは 350～400[kg]にカウントされることになる。これでは正しい評価が行えない。そこで、実測値と予測値の誤差を算出し、それをグラフ化した。その結果が図 5.5 である。

図 5.5 には誤差割合とデータ数の関係を示した。誤差割合とは以下(5.1)式に示すような

式で与えられた値を指している。

$$(\text{誤差割合}) = \frac{(\text{予測値}) - (\text{実測値})}{(\text{実測値})} \quad (5.1)$$

図 5.5 を見れば、 $-0.1 \sim 0$ と $0 \sim 0.1$ に多くのデータが集中していることが分かる。これは 10% 以内の誤差で予測ができていることを表しており、例えば実際の収穫量が 350kg であったとき、予測される収穫量が 315~385kg と算出されるということである。これはかなり良い結果が得られているのではないかと考えられる。しかし、誤差 10% 以内で予測できるデータは全データの 54.8% (219 データのうち 120 データ) で、残りの 46% 弱は誤差 10% 以上となる。

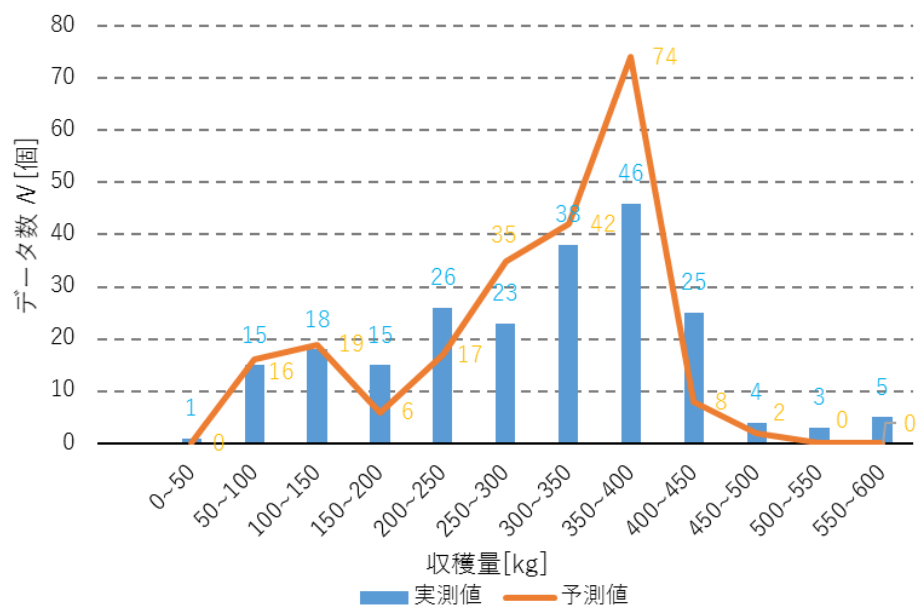


図 5.4：実際の収穫量と予測収穫量のデータちらばり

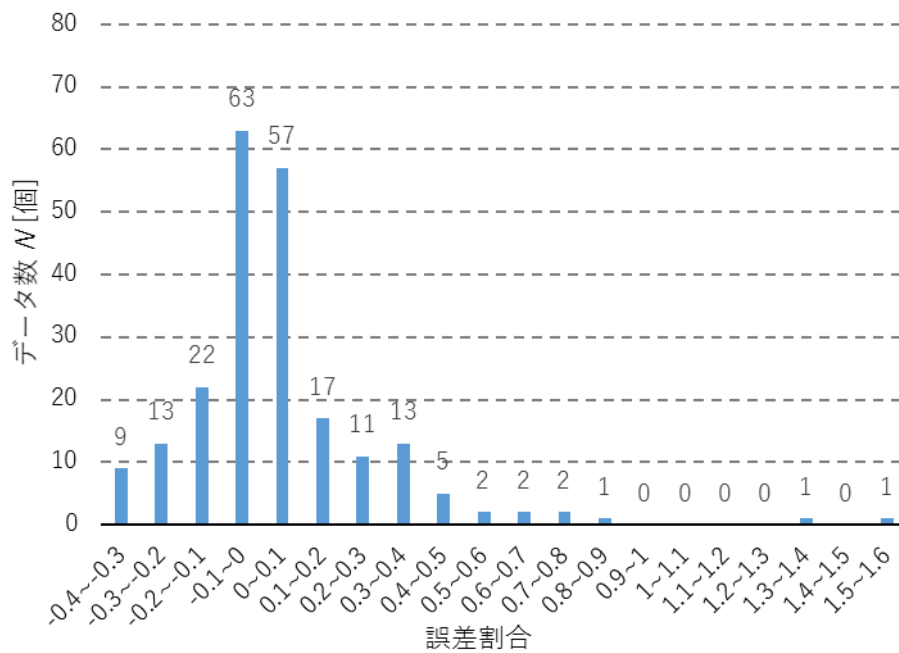


図 5.5 : 実際の収穫量と予測収穫量の誤差

この結果を JA 群馬担い手サポートセンターの方々にご覧いただいたところ、より精度の高いものにしてほしいとのことだった。そこでより高精度な予測を行っていくために、AI の導入を考えた。次章からは AI の導入を試みる。

6. 深層学習^{[6][7]}

4章では、重回帰分析を用いることで環境パラメータから収穫量を求める回帰式を算出してきた。様々な条件を試してみたが、最も良い結果は表 4.12 に示すものだった。さらに 5章では実際に収穫量の予測を行い、現場の方々に見ていただいたがあまり目を引くような結果にはならなかった。そこで、深層学習 AI を用いることによって更なる精度向上を目指していく。

6.1. ニューラルネットワークの構造

深層学習と一口に言っても様々な種類が存在し、何を求めるのかによってその手法を使い分けていかなければならないことが分かっている。我々の研究では、ハウス内の環境パラメータ（およそ 15 分間隔の時系列データ）を用いて、予想される収穫量を算出することを目標としている。環境パラメータが時系列データであること、キュウリは日々の環境の蓄積からなる産物であることから、再帰型ニューラルネットワーク（リカレントニューラルネットワーク、RNN とも言う。以降、RNN と表記する。）を用いて解析を行うことが妥当であると判断した。そのため、RNN を用いることでキュウリの収穫量を予測していく。

RNN の導入にあたって考えているニューラルネットワーク（以降、NN と表記する。）の構造について述べていく。まず、NN は大きく分けて入力層、中間層、出力層の 3 層に分かれており、特に中間層が 3 層以上あるネットワーク構造で計算を行うことを **Deep Learning** と呼んでいる。

さて、JA 群馬様よりいただいた環境パラメータおよび収量データを NN 構造のどこに割り当てるのかを図に表したものが図 6.1 である。これまで重回帰分析を行う中で、多重共線性と呼ばれるデータ同士の相関性が高いものが数パラメータほど存在した。重回帰分析、AI いずれにおいても統計学という点では変わらず、相関性が高いものを並列に入力しても深層学習が進まないのではないかと判断した。そこで多重共線性を含むパラメータは抜かず、厳選した 8 変数のみを入力層に与えた。そして、中間層では RNN による計算処理を行い、出力層で、予想される数日後の収穫量を算出する。

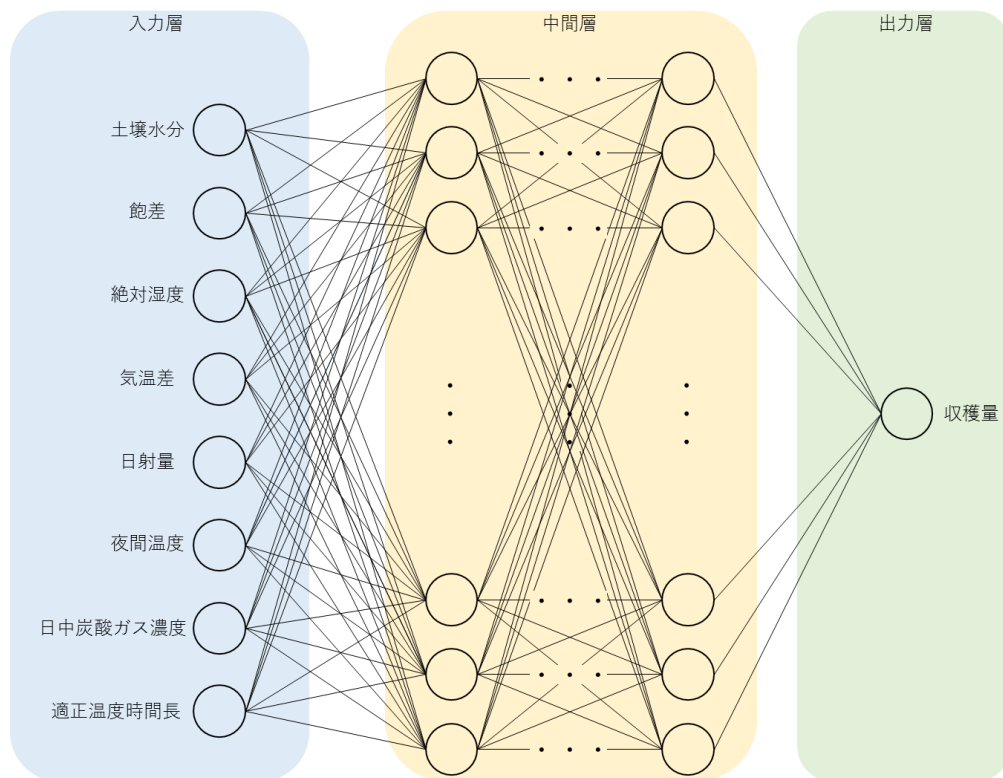


図 6.1 : ニューラルネットワーク (NN) の構造

6.2. RNN 学習における問題点

前節では、収量予測では RNN を用いることが妥当であるということ、入力層と出力層にはどのパラメータを置くのかを述べてきた。しかし、「入力層にはどのような形式のデータを、どのように入力するのか」「入力層と教師データの組はどのようにセットしていくのか」などといった具体的なデータ処理の方法を示すことはできなかった。というのも、JA 群馬様よりいただいているデータを AI の学習プログラムで扱うのには大きな問題が 2 つある。以下にその問題点を示す。

- ① 通常、AI の学習プログラムを構築する際には「学習データ」と「教師データ」の数を一致 (1:1 対応) させる必要があるが、所持しているデータ比はおおよそ $96 \times n : 1$ (15 分に 1 度の間隔でデータがあり、一日分で約 96 データ。植物の成長し収穫される過程を考えると 1 日の学習では不足するため n 日分必要。収穫量は 1 日 1 データ。) になってしまう。

- ② 教師データ 1 件に対して、学習データの数がまばらになってしまう。(必ずしも 15 分に 1 度の記録ではなく、16 分に 1 度、20 分に 1 度などの場合もあるため 1 日当たりのデータ数が日によって一致していない。)

この問題点を考慮した学習方法とするために新たなニューラルネットワークの構造を構築しなければならない。そのために必要な技術について、次項に示していく。

6.3. Truncated Backpropagation Through Time

①, ②を考慮した学習方法を構築するには、一般的な RNN の学習方法とは明らかに異なる。その決定的な違いは時系列データと教師データの比率が 1:1 対応していない点にある。本来、RNN の学習にはある時刻 t に対して学習データと教師データの両方が存在し、比率は 1:1 にする。しかし、我々が予測しようとしているキュウリの収穫量は何日分 (1 日で 96 データ) もの学習データに対して教師データが 1 データとなっているため、その比率は $n:1$ である。これを考慮して学習する方法として、Truncated Backpropagation Through Time (以降、Truncated BPTT と略す) という技術の応用を提案する。そもそも、Truncated BPTT とはどのような技術であるのか示した上でどのような応用をしていくのか本節では示していく。

6.3.1.RNN とは

Truncated BPTT について説明するためには RNN の基本構造を知っている必要があるため、RNN から順に説明を行っていく。

RNN の特徴はループする経路 (閉じた経路) を持つことである。このループする経路を持つことによって、データは絶えず循環することができる。そしてデータが循環することにより、過去の情報を記憶しながら最新のデータへと更新される。RNN で用いられるレイヤを「RNN レイヤ」という名前と呼ぶことにすると、RNN レイヤは図 6.2 のように書くことができる。図 6.2 に示すとおり、RNN レイヤはループする経路を持つ。このループする経路によって、データがレイヤ内を循環することができるようになる。図 6.2 では時刻を t として、 x_t を入力としている。これは時系列データとしてレイヤに入力されることを示している。そしてその入力に対応する形で h_t が出力される。

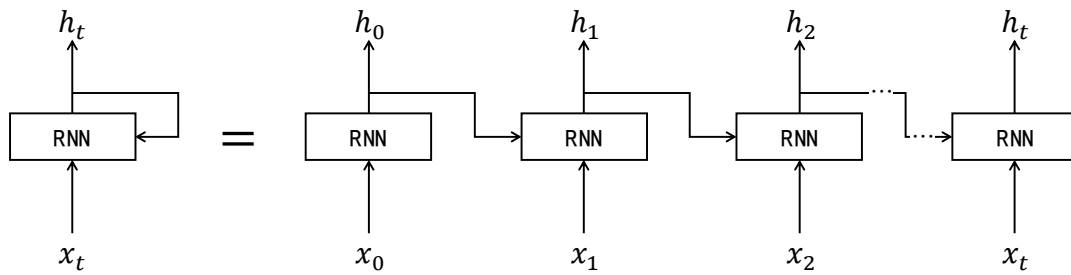


図 6.2 : RNN レイヤ

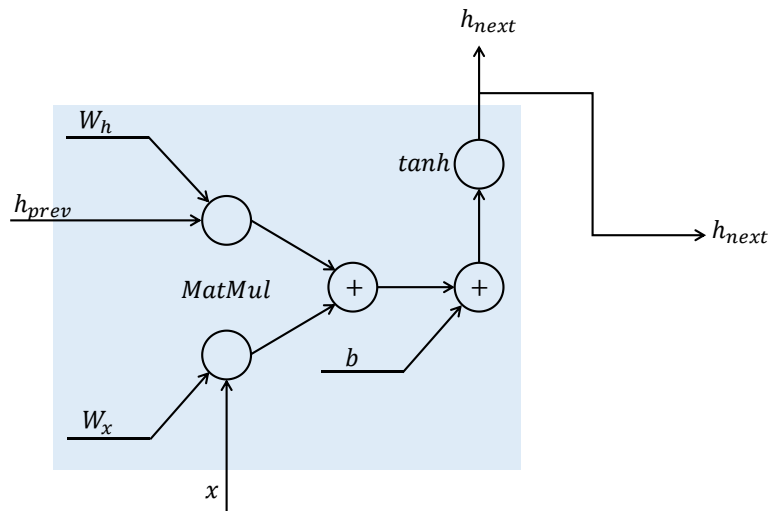


図 6.3 : RNN レイヤの内部構造

図 6.2 を見て分かる通り、各時刻の RNN レイヤはそのレイヤへの入力と 1 つ前の RNN レイヤからの出力を受け取る。そして、その 2 つの情報を元にその時刻の出力が計算される。このとき行う計算は以下に示す(6.1)式であり、模式的に図で表したものが図 6.3 である。

$$h_t = \tanh(h_{t-1}W_h + x_tW_x + b) \quad (6.1)$$

RNN では重みが 2 つあり、1 つは入力 x を出力 h に変換するための重み W_x 、もう 1 つは 1 つ前の RNN の出力を次時刻の出力に変換するための重み W_h である。また、バイアス b がある。

6.3.2. Backpropagation Through Time とは

図 6.4 に示す通り、ループを展開した後の RNN は誤差逆伝播法を使うことができる。つまり最初に順伝播を行い、続いて逆伝播を行うことで目的とする勾配を求めることができる。ここでの誤差逆伝播法は、「時間方向に展開したニューラルネットワークの誤差逆伝播

法」ということで **Backpropagation Through Time**, 略して **BPTT** と呼ぶ。この **BPTT** によって **RNN** の学習は行えるように見える。しかし、その前に解決しなければならない問題がある。それは、長い時系列データを学習する場合である。なぜそれが問題であるのかは時系列データの時間サイズが大きくなるに比例して、**BPTT** で消費するコンピュータの計算リソースも増加することになるからである。また、時間サイズが長くなると逆伝播時の勾配が不安定になることも問題である。これを解決する為に **Truncated BPTT** という技術がある。

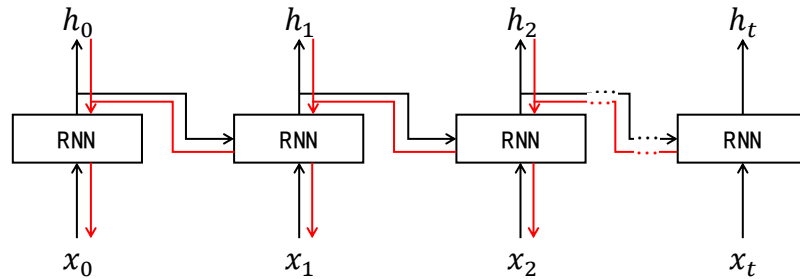


図 6.4 : RNN レイヤに対する誤差逆伝播法

6.3.3.Truncated BPTT とは

大きな時系列データを扱うときに通常用いられるのが、ネットワークのつながりを適当な長さで「断ち切る」ことである。これは、時間軸方向に長くなりすぎたネットワークを適当な長さに切り取ることで、小さなネットワークを複数作るというアイディアで、それらひとつひとつに対して誤差逆伝播法を行う。これが **Truncated BPTT** と呼ばれる手法である。

Truncated BPTT ではネットワークのつながりを断ち切るが、正しくはネットワークの「逆伝播」のつながりだけを断ち切るということである。すなわち、順伝播のつながりは維持されたままになるのである。一方、逆伝播のつながりは適当な長さで切り取り、その切り取られたネットワーク単位で学習を行う。

ここからは具体的に例を挙げて説明していく。例えば、1000 個の時系列データがあったとする。この時系列データを扱うとき **RNN** レイヤを展開すると、横方向に 1000 個のレイヤが並んだネットワークになる。もちろん、どれだけレイヤが並んだとしても誤差逆伝播法によって勾配を計算することは可能である。しかしそれがあまりにも長いと、計算量やメモリの使用量などの点で問題になる。またレイヤが長くなるに従い、勾配が徐々に小さくなることもあり、勾配が前時刻へと届かなくなる。そこで図 6.5 に示すように横方向に長く伸びたネットワークの逆伝播のつながりを適当な長さに断ち切ることを考える。このように逆伝播のつながりを切ってしまうと、それより未来のデータについて考える必要がなくなり、ブロック単位で誤差逆伝播法を完結することができるようになる。

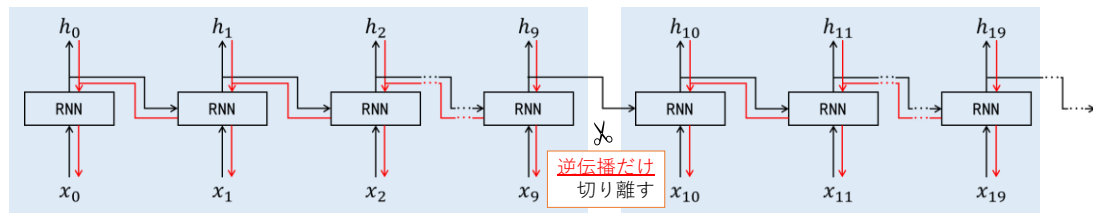


図 6.5 : 逆伝播のつながりを切断

6.4. 実装

前節で述べてきた **Truncated BPTT** を応用することによって RNN を実装していくことにする。図 6.4 に示したように、入力データ x_t を RNN 処理に通すと出力データ h_t を出すのがセオリーである。しかし、本稿での実装における h_t は次の入力データと共に再度 RNN 処理に通すようにする。これを繰り返し、つながりを切断した部分まで出力データを記録しないようにする。(図 6.6) こうすることで、複数の入力データ（時系列データ）に対して 1 つの出力データで済ませることができる。

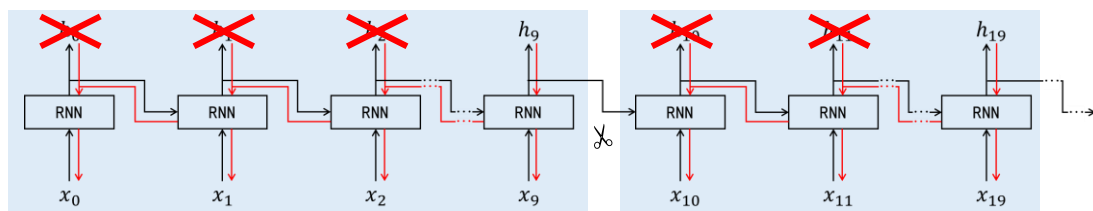


図 6.6 : 出力データを 1 つのつながりでひとつにする

6.4.1. 実装の流れ

前節までで述べてきた **Truncated BPTT** は、RNN 層の部分のみを見て話を進めてきたが、AI の学習においては RNN 層のみを見ているだけでは実装できない。それでは RNN 層以外にどのような層が存在しているのか。これを図示したものが図 6.7 になる。

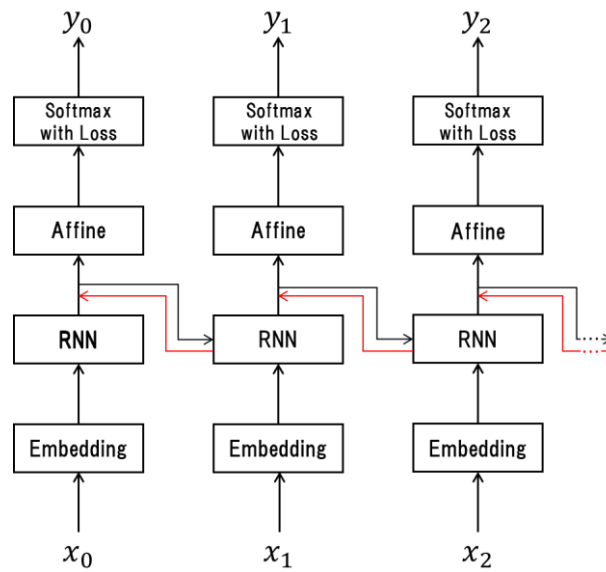
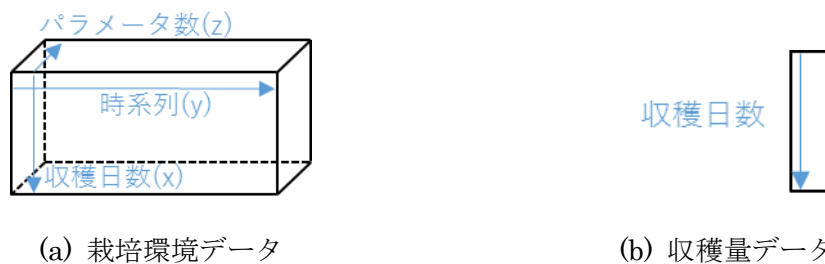


図 6.7 : RNN ネットワーク図

Embedding 層ではデータ入力のための下準備を行う層である。そして、RNN 層は上記に示したような学習のための層、Affine 層は隣接する全てのニューロン間の結合を行うための層、Softmax 層は Affine 層によって出力されたデータを確率に変換する層になる。このように各層で処理を加えられることによって、データは様々な変化を起こし学習が進んでいく。実際の実装もこのような流れで構築を行っていく。次項以降に各層の実装内容を示していく。

6.4.2. データセット

RNN のデータセットにはキュウリの栽培環境（気温や湿度といった各種パラメータ）と収穫量[kg]を用いる。しかし、栽培環境データと収穫量データの数合わないため、データセットに工夫する必要がある。本稿ではセット方法を以下のようにした。



(a) 栽培環境データ

(b) 収穫量データ

図 6.8 : データセット形式

図 6.8 に示すように栽培環境データは 3 次元配列とし、1 次元目 (x 軸) を収穫日数、2 次元目 (y 軸) を時系列、3 次元目 (z 軸) をパラメータ数とした。また、収量データは収

穫量のみを持つ1次元配列とした。このとき栽培環境データの1次元目と収量データの要素数は揃えることにする。

6.4.3. ミニバッチの作成

データセットから無作為にデータを選び出す。この際、収穫日数ごとに（yz平面は固まりとして）選び出す。（図 6.9）ミニバッチ数の指定は後述する。

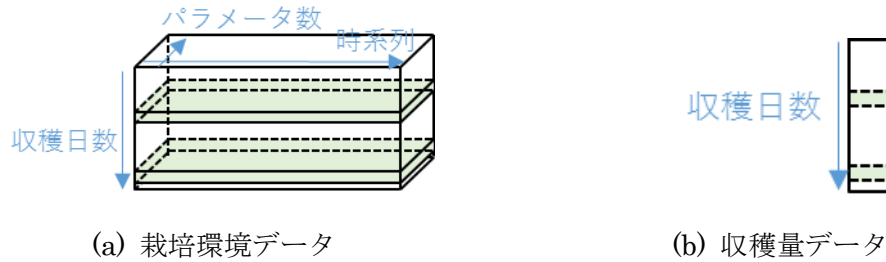


図 6.9 : ミニバッチの作成

6.4.4. RNN 順伝播

この層では自己回帰型の構造を持たせ、前の情報を取り入れた解析を行っていく。前項で抜き出した学習セットを時系列方向に1データずつ切り取りながら計算を行っていく。

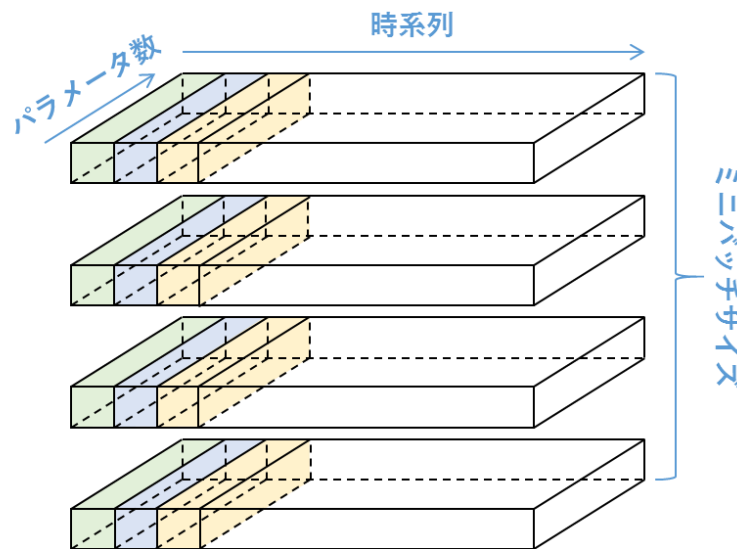


図 6.10 : RNN 順伝播の計算

ミニバッチサイズを m 、時系列データ数を n 、パラメータ変数を o とおけば、RNN 層に渡される配列は $(m \times n \times o)$ の3次元である。これを図 6.10 のように時系列方向に1データずつ切り取りながら計算を行っていく。切り取ったデータは $(m \times o)$ の2次元配列となり、あらかじめ用意しておいた入力に対する重みや隠れ状態ベクトル、バイアス等と共に(6.2)式

のような計算がなされ出力となる。

$$h_t = \tanh(x \cdot W_x + h_{t-1} \cdot W_h + b) \quad (6.2)$$

(6.2)式における x は切り取った $(m \times o)$ の2次元データ、 W_x は入力データの重み ($(o \times p)$ の2次元配列データ)、 h_{t-1} は1つ前の時系列における計算結果 (以後、隠れ状態ベクトルと呼ぶ、初期値は「Xavier の初期値」を使用)、 W_h は隠れ状態ベクトルに対する重み、 b はバイアス (初期値は0) である。

(6.2)式を時系列データ分繰り返し計算することで隠れ状態ベクトル h_t は n セットでき、それらを結合することにより $h_s = (m \times n \times p)$ (p は任意の値) の3次元データが完成する。

6.4.5.Affine 順伝播

ここまではミニバッチ単位で計算を行ってきたが、学習においては汎用性を出さなければならない。そこで、ミニバッチという固まりごとの考え方を一度バラしていく。

図 6.11 のように RNN 順伝播で算出した3次元データ $h_s = (m \times n \times p)$ を $h_{shape} = (mn \times p)$ の2次元データに変換した。その後 RNN 順伝播と同様、(6.3)式のような計算を行い全結合する。

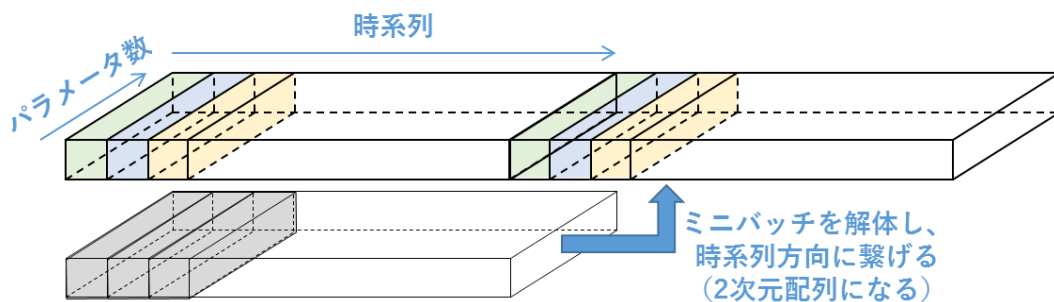


図 6.11 : Affine 順伝播の構成

$$out = h_{shape} \cdot W + b \quad (6.3)$$

(6.3)式における W 、 b はそれぞれ重み ($(p \times q)$ の2次元データ)、バイアスを示す。この式により全結合された結果は、 $(m \times n \times q)$ の3次元データに戻る。ここで、 q の値は収穫量の予測のレンジを示すことになる。例えば、 q の値を1000とすれば、収穫量を0～999kgの範囲で予測できるということである。

6.4.6. Time Softmax with Loss 順伝播

前項までに計算を行ってきた結果と教師データを照らし合わせることで、Loss を求めていく。ここでデータセットの時系列データと教師データの数が一致していないという問題が生じる。これを解決するために、図 6.12 に示すような方法を取り入れる。

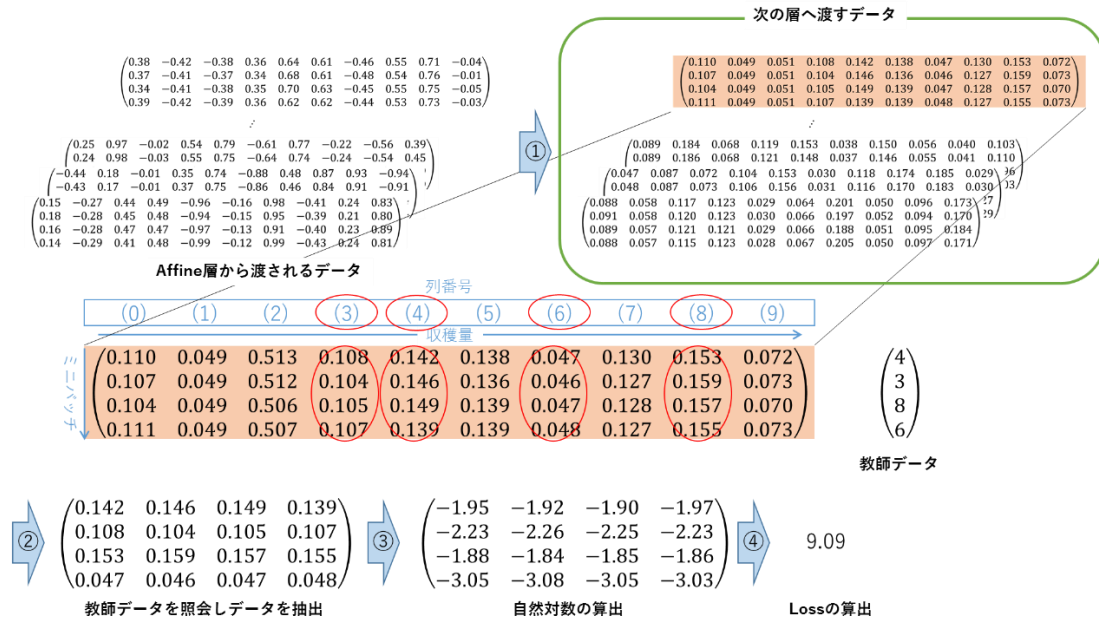


図 6.12 : Loss 算出までのデータ処理

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)} \quad (6.4)$$

前の層である Affine 順伝播層から渡されたデータは、①の工程で Softmax 関数((6.4)式)によって計算され、行ごとに列を全て足すと 1 になるように値を置換していく。次に②の工程では、時系列データの最後（最新のデータ）のみを教師データと照らし合わせながら抽出を行っていく。③では、②で抽出した行列の自然対数をとる。最後に④では③で算出した自然対数の和を求め、ミニバッチ数で割った値の絶対値をとったものである。

こうして最後に算出された Loss はニューラルネットワークの学習精度を表し、0 に近ければ近いほど、学習はうまくいっているということになる。また、次の層へ渡すデータは図 6.12 中緑枠で示した部分である。

6.4.7. Time Softmax with Loss 逆伝播

逆伝播では、順伝播で行ってきた計算を逆方向に伝播させ、勾配（微分）を求めていくことで、これまでに登場した重み w やバイアス b の更新を行っていく。このようにすることで計算により予測した収穫量と既知の収穫量（教師データ）の値を近づけていくことができるようになる。

それでは本項のタイトルでもある Time Softmax with Loss の逆伝播について説明していく。図 6.12 中緑枠で示した配列の教師データの収穫量に該当する要素から 1 を引く。このような計算を行うことで Time Softmax with Loss の逆伝播が完成する。

6.4.8. Affine 逆伝播

Time Softmax with Loss の逆伝播で算出した $(m \times q)$ の 2 次元配列（以降 out と呼ぶ）と Affine の順伝播で算出した 3 次元配列を用いることにより、勾配を求めていく。ここで問題となるのは out が時系列方向には 1 データしかなく、Affine 順伝播の出力である 3 次元行列 $((m \times n \times q)$ の配列) とは形が揃っていない点である。この問題を解決する為に、 out と同様、時系列の最後尾データのみを残すことで形をそろえる（以降 rx と呼ぶ）。形をそろえることで勾配を求められるようになる。

上述した out および rx を用いることで、重みの勾配、バイアスの勾配、Affine 順伝播の入力の勾配を求めていく。重みの逆伝播経路は図 6.13 中緑線であるため、勾配は out と入力 x の内積となる。同様に、重みの勾配は out 、Affine の入力の勾配は $out \cdot W$ となる。

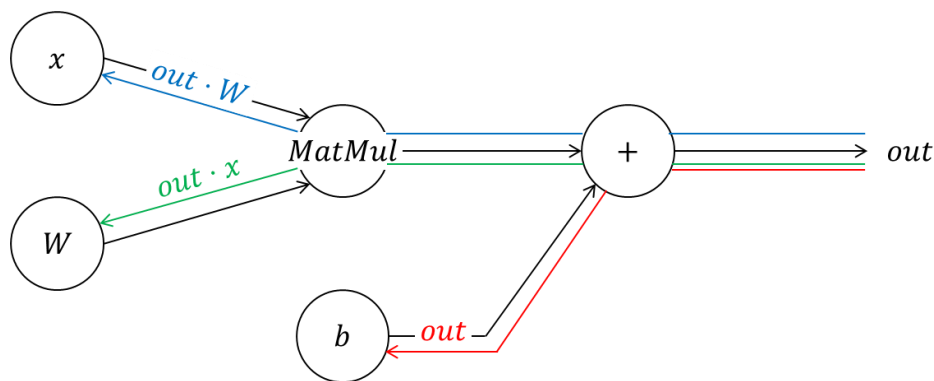


図 6.13 : Affine の逆伝播

6.4.9.RNN 逆伝播

Affine 順伝播の入力勾配を用いて本項では RNN の逆伝播を行っていく。RNN の逆伝播も Affine 逆伝播と同様に図 6.14 のような伝播経路を通して勾配を求めていく。厳密には RNN の順伝播では \tanh を用いているため \tanh の逆伝播も行っているが、図の簡易化のため、図 6.14 では割愛した。

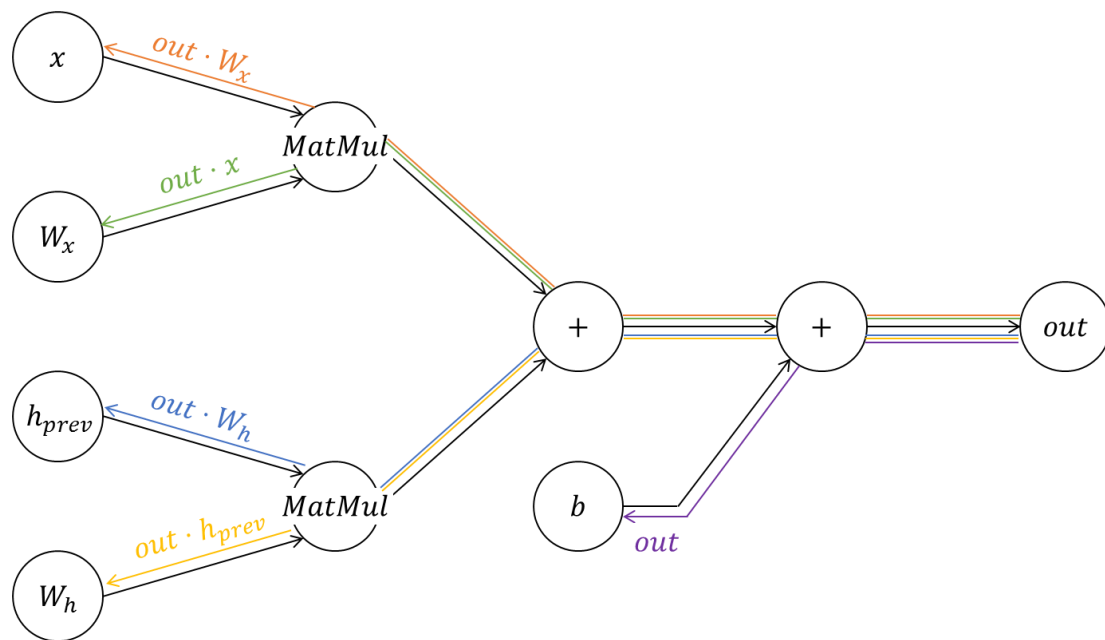


図 6.14 : RNN の逆伝播

6.4.10. 最適化 Optimizer

最後に最適化 (optimize) を行う。ここで最適化を行うことによって、より高い学習精度が得られることになる。最適化とは上述してきた重みの勾配をある関数によって解析し、予測に適した重みに更新していくことを指す。この最適化には様々な関数が存在するが (例えば SGD, Adam, AdaGrad など)、本節では RNN 実装における手順をメインに記述してきているため、それらの紹介は割愛する。

6.5. 機械学習

6.4 節のような手順で RNN を構築し学習を行っていく。その学習におけるパラメータ及び結果を本節で示す。

6.5.1. 各種パラメータ

実装に際して、かなり多くの変数が現れてきた。本項ではそれらのパラメータの値を一覧として示す。

表 5.1：各種パラメータとその説明

| 適用レイヤ | パラメータ名 | 値 | 説明 |
|------------|-----------------|----------|-------------------------------------|
| データセット | Data_per_day | 97 | 1 日当たりの時系列データ数 |
| | Days_RANGE | 10 | 教師データ 1 件当たりの時系列日数 |
| | PREDICT_RANGE | 7 | 何日後の収穫量を予測するか |
| | PARAMETER | 8 | 環境データ変数 |
| ミニバッチ | MINI_BATCH_SIZE | 20 | ミニバッチサイズ |
| RNN 順伝播 | 重み W_x | (8,1) | 重みの配列サイズ 初期値は Xavier を用いた |
| | 重み W_h | (1,1) | |
| | バイアス b | 0 | バイアスの初期値 |
| Affine 順伝播 | 重み W | (1,1000) | 重みの配列サイズ 初期値は Xavier を用いた |
| | バイアス b | 0 | バイアスの初期値 |
| optimizer | 最適化関数 | SGD | 最適化に用いる関数 |
| | 学習係数 lr | 0.1 | |
| - | EPOCH | 1000 | 学習回数 |

以上が各種パラメータになっている。赤字で示した値および関数は容易に変更が行える。今回の学習には表 6.1 のように設定し学習を行う。

6.5.2. 学習結果

上述してきた手法で学習を行った。その結果、Loss および Perplexity は次のようになった。

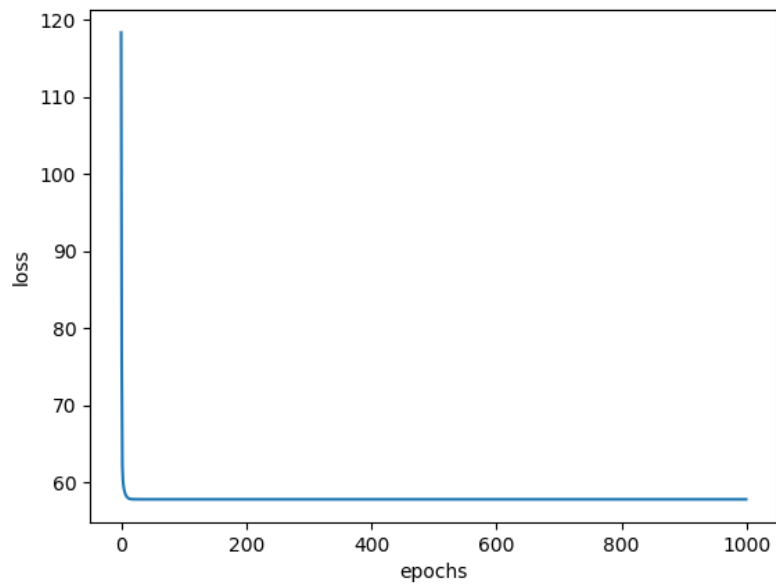


図 6.15 : Loss の推移

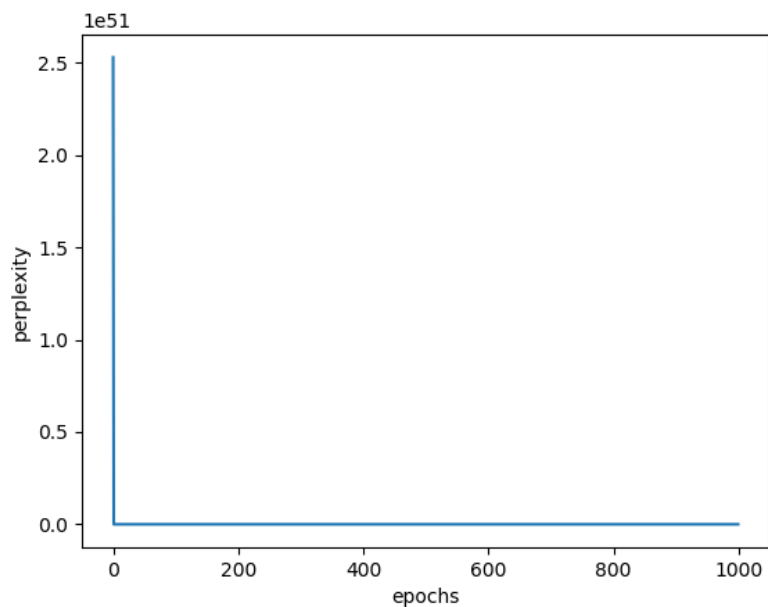


図 6.16 : Perplexity の推移

図 6.15 の Loss の推移は 58 ほどで落ち着き、これ以上は落ちることがなかった。学習がきちんと行えていれば、Loss の値は 0 に限りなく近づいていくものであるが、そうならないため、学習が適切には行えていないと読み取ることができる。図 6.16 には

Perplexity と呼ばれる評価方法を用いた。Perplexity とは分岐数あるいは選択肢の数を表したものであり、確率の逆数で定義される。つまり、確率が 0.05 であると、Perplexity は 20 と表され、予測の選択肢が 20 通りあるということを示す。改めて図 6.16 の波形を見ると、一見 Perplexity の値が 0 に近づいており、予測できているように感じられるがレンジが 10^{51} になっているため、この波形では予測できているとはいいがたい。

7. 結論

本稿では、きゅうりの栽培環境の最適値を求めることを目的として研究を進めてきた。

そのために、4章では重回帰分析と呼ばれる多変量解析の1種を用いて解析に必要な環境パラメータを厳選し、それらを用いて環境パラメータと収穫量の関係式(回帰式)を算出した。これにより、どの環境パラメータが収穫量に大きく関わっているのかを示すことができた。しかし、この回帰式の評価値となる決定係数や自由度修正済み決定係数の値は0.6ほどにとどまってしまい、信頼性を考えると不安の残る結果となってしまった。

5章では、4章で扱った技術を応用することで収穫量の予測を行った。全データの54.8%は10%以内の誤差で予測できたが、そのほかは10%以上の誤差となり、実用化するにはまだ課題の多い結果となってしまった。

6章では5章での課題となった信頼性をより向上させるために深層学習AIを用いた。従来の深層学習ライブラリであるTensorFlowやChainerなどでは、我々が行いたいデータセットの形を学習させることができない為、ライブラリを用いない学習アルゴリズムを考案し実装した。しかし、学習が思うように進まなかった。

今後の課題として、学習アルゴリズムの改良や表6.1に示した赤字のパラメータの最適値を見つけ、Lossを0により近くなるような学習を行っていきたいと考えている。更に、学習できた結果を用いて収穫量予測を行いきゅうり栽培における好条件を見つけ出していきたい。

謝辞

本研究を進めるにあたり、3年間熱心に指導してくださいました本島 邦行 教授に深く感謝の意を表すとともに、厚く御礼申し上げます。また、私の修士論文の審査を引き受けてくださいました小林 春夫 教授、弓仲 康史 教授に心から感謝申し上げます。さらには、環境データのご提供および様々なご教示をくださいました JA 群馬担い手サポートセンターの金井さん、山上さん、今井さんをはじめとする皆さんに心より感謝申し上げます。

最後に、研究室生活でお世話になりました先輩、同期、後輩、全ての方々に感謝の意を表し、謝辞とさせていただきます。

本研究の一部は、令和元年度群馬大学食健康科学に関する地域連携研究助成を受けたものであることを付記し、関係各位に心より感謝致します。

参考文献

— 第 1 部 —

- [1] 加藤丈和, 湯浅健史, 松山隆司:「オンデマンド型電力制御システム」, 情報処理学会誌論文, Vol.54, No.3, pp.1185-1198(2013)
- [2] 環境省:「地球温暖化対策計画」, <https://www.env.go.jp/press/files/jp/102816.pdf>
- [3] 東京電力エナジーパートナー株式会社:「スマートメーター」,
<http://www.tepco.co.jp/ep/private/smartlife/smartmeter.html> (参照 2020-01-10)
- [4] 国立研究開発法人情報通信研究機構:「スマートメーター用通信標準規格『ECHONET Lite』及び『Wi-SUN』を搭載した小型・省電力“無線機”の開発に成功」,
<https://www.nict.go.jp/press/2013/05/23-1.html> (参照 2020-01-10)
- [5] 八木田克英, 岩船由美子:「HEMS 普及のための消費者受容性に関する検討」, 第 23 回日本エネルギー学会大会, 6-2-4
- [6] 阿久澤一起, 高橋昌史, 本島邦行, 羽賀望:「近距離無線通信技術を応用した住宅向けマイクログリッドの構築」, 平成 27 年度第 5 回電気学会東京支部神奈川支所研究発表会, KNG-16-013, p.35
- [7] 高橋昌史, 本島邦行, 羽賀望:「IoT を応用した電力デマンド制御システム」, 平成 28 年電気学会産業応用部門大会, PB-1, 5-22
- [8] Panasonic:「HEMS で何ができるの?」,
<http://sumai.panasonic.jp/hems/ability.html> (参照 2020-01-10)
- [9] 東山, 宮岡, 谷口, 佐藤:「触覚と痛み」, プレーン出版(2005)
- [10] ROHM:「Wi-SUN | Wi-SUN とは?」, (参照 2020-01-14)
https://www.rohm.co.jp/electronics-basics/wireless/wireless_what4
- [11] 高橋昌史:「IoT を応用した電力デマンド制御システム」, 平成 29 年度修士論文
- [12] 東京電力ホールディングス株式会社:「TEPCO 統合報告書 2019」,
http://www.tepco.co.jp/about/ir/library/annual_report/pdf/201910tougou-j.pdf
- [13] Robert Faludi:「XBee で作るワイヤレスネットワーク」, オーム社(2015)
- [14] 鄭立:「ZigBee 開発ハンドブック」, 株式会社リックテレコム(2012)
- [15] 水澤一太:「IoT を用いた家庭内電力デマンド制御法」, 平成 29 年度学士学位論文

— 第2部 —

- [1] JA 全農ぐんま HP : <https://www.gm.zennoh.or.jp/index.php>
- [2] 栗原伸一：「入門統計学—検定から多変量解析・実験計画法まで」，オーム社(2011)
- [3] 石井俊全：「意味がわかる多変量解析」，ベレ出版(2014)
- [4] 日本理学療法士学会：「多重共線性 multicollinearity」，
http://jspt.japanpt.or.jp/ebpt_glossary/multicollinearity.html
- [5] 株式会社 Albert：「重回帰分析とは | データ分析基礎知識」，
https://www.albert2005.co.jp/knowledge/statistics_analysis/multivariate_analysis/multiple_regression
- [6] 斎藤康毅：「ゼロから作る Deep Learning
—Python で学ぶディープラーニングの理論と実装」，O'REILLY(2018)
- [7] 斎藤康毅：「ゼロから作る Deep Learning②—自然言語処理編」，O'REILLY(2018)

研究業績

論文

- [1] 水澤一太，高橋昌史，本島邦行：「家電独自判断による家庭内電力デマンド制御法」，
電気学会論文誌 D（産業応用部門誌），Vol.139 No.3 pp.316-321

国内学会発表

- [1] 水澤一太，本島邦行：「家電独自判断による家庭内電力デマンド制御法」，第 9 回電気
学会群馬・栃木支所合同研究発表会，ETG-19-15，ETT-19-15